Summer 2001

Designing web-based adaptive learning environment : distils as an example

Lilian Cao
New Jersey Institute of Technology

ABSTRACT

DESIGNING WEB-BASED ADAPTIVE LEARNING ENVIRONMENT: DISTILS AS AN EXAMPLE

by
Lilian Cao

In this study, two components are developed for the Web-based adaptive learning: an on-line Intelligent Tutoring Tool (ITT) and an Adaptive Lecture Guidance (ALG). The ITT provides students timely problem-solving help in a dynamic Web environment. The ALG prevents students from being disoriented when a new domain is presented using Web technology. A prototype, Distributed Intelligent Learning System (DISTILS), has been implemented in a general chemistry laboratory domain.

In DISTILS, students interact with the ITT through a Web browser. When a student selects a problem, the problem is formatted and displayed in the user interface for the student to solve. On the other side, the ITT begins to solve the problem simultaneously. The student can then request help from the ITT through the interface. The ITT interacts with the student, verifying those solution activities in an ascending order of the student knowledge status. In DISTILS, a Web page is associated with a HTML Learning Model (HLM) to describe its knowledge content. The ALG extracts the HLM, collects the status of students' knowledge in HLM, and presents a knowledge map illustrating where the student is, how much proficiency he/she already has and where he/she is encouraged to explore. In this way, the ALG helps students to navigate the Web-based course material, protecting them from being disoriented and giving them guidance in need.

Both the ITT and ALG components are developed under a generic Common Object Request Broker Architecture (CORBA)-driven framework. Under this framework, knowledge objects model domain expertise, a student modeler assesses student's knowledge progress, an instruction engine includes two tutoring components, such as the ITT and the ALG, and the CORBA-compatible middleware serves as the communication infrastructure. The advantage of such a framework is that it promotes the development of modular and reusable intelligent educational objects. In DISTILS, a collection of knowledge objects were developed under CORBA to model general chemistry laboratory domain expertise. It was shown that these objects can be easily assembled in a plug-and-play manner to produce several exercises for different laboratory experiments. Given the platform independence of CORBA, tutoring objects developed under such a framework have the potential to be easily reused in different applications.

Preliminary results showed that DISTILS effectively enhanced learning in Web environment. Three high school students and twenty-two NJIT students participated in the evaluation of DISTILS. In the final quiz of seven questions, the average correct answers of the students who studied in a Web environment with DISTILS (DISTILS Group) was 5.3, and the average correct answers of those who studied in the same Web environment without DISTILS (NoDISTILS Group) was 2.75. A t-test conducted on this small sample showed that the DISTILS group students significantly scored better than the NoDISTILS group students.

DESIGNING WEB-BASED ADAPTIVE LEARNING ENVIRONMENT: DISTILS AS AN EXAMPLE

by
Lilian Cao

A Dissertation
Submitted to the Faculty of
New Jersey Institute of Technology
In Partial Fulfillment of the Requirements for the Degree of
Doctor of Philosophy in Industrial Engineering

Department of Industrial and Manufacturing Engineering

May 2001

APPROVAL PAGE

DESIGNING WEB-BASED ADAPTIVE LEARNING ENVIRONMENT: DISTILS AS AN EXAMPLE

Lilian Cao

Dr. Gülgen Bengu, Dissertation Advisor DATE
Associate Professor of Industrial and Manufacturing Engineering, NJIT

Dr. Jack Gelfand, Committee Member DATE
Research Scientist of Psychology, Princeton University

Dr. One-Jang Jeng, Committee Member DATE
Assistant Professor of Industrial and Manufacturing Engineering, NJIT

Dr. Barbara Kebbekus, Committee Member DATE
Professor of Chemical Engineering, Chemistry, and Environmental Science, NJIT

Dr. Arijit Sengupta, Committee Member DATE
Associate Professor of Manufacturing Engineering Technology, NJIT

BIOGRAPHICAL SKETCH

Author :	Lilian Cao
Degree:	Doctor of Philosophy
Date:	May 2001

Undergraduate and Graduate Education

- Doctor of Philosophy in Industrial Engineering,
  New Jersey Institute of Technology, Newark, NJ, 2001

- Master of Science in Automation,
  Tsinghua University, Beijing, China, 1995

- Bachelor of Science in Industrial Automation,
  Huazhong University of Science and Technology, Wuhan, China, 1992

Major: Industrial Engineering

Publications:

Bengu G., Kebbekus B., Cao L., "Reengineering the General Chemistry Laboratory Experience: Use of an Intelligent Learning System", *Sharing the Future –II A Working Conference by NSF-Succeed/Gateway/Foundation*, Clemson. S.C., March, 2001.

Bengu G., Liu C., Tang X., Erkavun G., Zhang W., Cao L., "A Web based Courseware Preparation Guide Using a Syllabus Based Teaching/Learning Framework: for an Engineering Statistics Courseware", *ASEE Middle Atlantic Section Spring Conference*, 2001.

Cao L., Bengu G, "Web-based Agents for Reengineering General Chemistry Laboratory Education", *Journal of Educational Computing Research,* 24(3), 2000

Cao L., Li F., Chai Y., "A Partitioned Matrix-based Model for Material Requirement Planning", *China High-Tech Letters,* 1997

Cao L., Li F, "Object-Oriented Modeling and Design for Bill of Material", *China Journal of Information and Control,* 1996

This dissertation is dedicated to my parents.

ACKNOWLEDGMENT

The author wishes to express his sincere gratitude to his supervisor, Dr. Bengu, for introducing me to this research and her consistent guidance and support throughout this research.

Special thanks to Dr. Jeng, Dr. Gelfand, Dr. Kebbekus, and Dr. Sengupta for serving as members of the committee. The author appreciates their insight and suggestions during this research.

Dr. Kluiber, Professor of the Chemistry Department at Rutgers University-Newark, provided expert assistance on the Chemistry content and assessment for the prototype system.

Thanks to NSF and New Jersey Center of Multimedia Research (NJCMR) for funding and supporting this research.

And finally, a thank you to Ping Cheng, Ritesh, and Guangyu Liu for their efforts on this project.

TABLE OF CONTENTS

TABLE OF CONTENTS
(Continued)

LIST OF TABLES

LIST OF FIGURES

LIST OF FIGURES
(Continued)

GLOSSARY

ALG	Adaptive Learning Guidance
BOA	Basic Object Adaptor
CAI	Computer Assisted Instruction
CORBA	Common Object Request Broker Architecture
DISTILS	DISTributed Intelligent Learning System
HLM	HTML Learning Model
HTML	Hyper Text Markup Language
IDL	Interface Definition Language
IIOP	Internet Inter-ORB Protocol
ITS	Intelligent Tutoring System
ITT	Intelligent Tutoring Tool
NJCMR	New Jersey Center of Multimedia Research
ORB	Object Request Broker
POA	Portable Object Adaptor
WAL	Web-based Adaptive Learning
WBL	Web Based Learning

CHAPTER 1

INTRODUCTION

1.1 Motivation

Today the world is characterized both by rapid technological progress and by major changes in the marketplace (Eidgahy 1998). In such a swiftly changing environment, life long learning is becoming a part of life. Education is critical for everyone to update and advance their technological skills to keep pace with industrial and technological progress. More and more people are returning for continuing education after years of work. Marsh (1999) reports that the U.S. college population is becoming older, and only 35% of currently enrolled students across the nation are below the age of 25. Furthermore, students are increasingly involved in part-time rather than full-time programs, financially unable to study in residence, and seeking flexibility and convenience in college course offerings.

The education needs of all the people in and out of U.S. are paramount. However, today's educational system is far from matching these needs. First, there is a serious shortage of faculty in most of branches of engineering (NSF 1992). Second, the demands of dynamic, just in time instruction or training in the near future cannot be attainable by current college educational methodologies and technologies. Therefore, it is necessary to seek approaches to enrich the education pipeline to address the demands of the future.

Web Based Learning (WBL) appears to have many advantages. First, powerful personal computers are very popular and widely available. Second, the Internet greatly reduces the limits of geographical distance. Over the Internet, E-mail, groupware, and bulletin boards provide convenient ways to communicate over distance. Educational content retrieval and communication are no longer confined to the traditional spaces of laboratories, schools, and libraries. Third, the Web makes it possible to run distributed interactive educational multimedia applications using a variety of geographically dispersed sources of information.

There are many kinds of WBL applications today (Bengu 1995; Bengu and Swarts 1996; Brooks 1997; Kortemeyer 1998; Li 1998; Marsh 1999; Paterson 1999; ST-Pierre et al 1999; Turoff 1999; Pincipe et al 2000; Cao and Bengu 2000). As a matter of fact, today most U.S. universities are taking advantage of the Web to deliver distance instruction. Paterson (1999) described the design of an undergraduate environmental engineering course in atmospheric physics and chemistry using Internet-based tools at Michigan Technological University. In this work, instructional tools include PDF partial notes, an electronic forum, Internet-based homework assignments, and term projects. Students' reviews of this class were overwhelmingly positive. Paterson concluded, based on student evaluations, that a traditional class, supplemented with appropriately designed Internet-based learning tools can yield a educational system that is more balanced among the various learning styles.

Much effort has also been put into Web materials at New Jersey Institute of Technology (NJIT) to assist education. Bengu (1995) developed an interactive multimedia courseware on manufacturing processes & systems by taking advantage of Web multimedia simulation technology. At NJIT, distance learning classes are usually delivered through videos and Web-based class site, which includes notes, homework assignments, projects, threaded discussion groups and exams. These courses were supervised by the instructor. As Turoff (1999) reported, students enjoyed learning in this way and some students in traditional on-site classes even took advantage of similar distance learning classes.

The college of education at the University of Alabama offers collection of courses via Web education, such as “Technology in Education” (Marsh 1999). Much of the course material is presented in this site for students' learning or reference. Marsh (1999) made the following observation: "There is overwhelming evidence that technologically delivered instruction--synchronous (e.g., radio, television) and asynchronous (i.e., film, videotape, screen-cam lectures, CD-ROM's, web-based instruction) --is equivalent to traditional instruction." Marsh (1999) noted that computer-delivered instruction can furnish students with far greater interaction than is possible in a large classroom or a small class where the lecture is the chief means of interaction.

Although Web-based learning is promising, it also brings new learning/teaching challenges. For example, reading from a computer screen inhibits students from using significant information that is readily available to them on paper (Kozma 1991). Writing notes, making comments and adding bookmarks are very useful activities for students to organize their own understandings and facilitate reading comprehension. These activities can be done easily with paper material, but are very difficult in today's Web based material. The Web is essentially a nonlinear medium. It uses non-sequential links to organize information. It is reported that learning in such an environment may burden students with cognitive load and disrupt their concentration in some situations, especially when an unfamiliar knowledge domain is presented (Gray 1993; Ashani 1998). Students in Web-based courses often work at odd hours, on highly variable schedules and at remote sites (Forbus 1998). It may be difficult to get timely help from the instructor or peer students, which is important for effective and active development of mental models in unfamiliar domains. For example, at NJIT, many students who attended the distance learning class – client server computing—often found difficulty in finishing the homework assignments. They could not get timely help from peer students or from the lecturer by email or the threaded forum. Most answers through email or threaded forum are often delayed or not very specific. Usually for the lecturer or peer students to review problems, they need to see the solution on-spot or they have to go over a time-consuming set up process.

Developing Web-based adaptive learning environment would effectively meet these challenges. In this study, we focus on designing a Web-based adaptive learning environment, and presenting a generic framework for facilitating its development. A prototype, DISTributed Intelligent Learning System (DISTILS), is developed and evaluated as a learning tool in general chemistry pre-laboratory domain.

1.2 Application Background and Objective

This study was motivated by the need to reengineer general chemistry laboratory education at NJIT. Laboratory time is always expensive. In this course, to ensure the least use of laboratory time and maximize the laboratory experience, students are required to demonstrate their understanding of the materials and procedures before they enter the actual laboratory. Previously, students were asked to review a traditional printed lab manual and fill in the pre-lab quiz sheet. Recently, the quiz sheet has been replaced with a computer-based (Fortran IV) pre-lab quiz system developed at Rutgers-Newark

Chemistry Laboratory (Kluiber 1996) and is used at the Departments of Chemical Engineering, Chemistry, NJIT and Chemistry Rutgers-Newark. Despite these efforts, students were having difficulty with this system and struggling to understand chemistry concepts underlying the laboratory experiments and analysis. Based on the experience of instructors, these results were partly attributed to not having enough lab assistants to help the students individually. Due to large number of students required to take these lab experiments, any improvement in the area of an individualized assistance in the pre-lab courses can provide significant advantages.

To address the above challenges, a comprehensive multimedia courseware has been prepared (http://bengu-pc2.njit.edu/trp-chem). This courseware includes twelve experiments as follows:

1. MEASURING THE DENSITY OF A SOLID AND A LIQUID
2. SOME NON-METALS ANDTHEIR COMPOUNDS
3. WATER OF HYDRATION
4. THE SOLVAY PROCESS
5. CALORIMETRY: EXPERIMENTS BASED ON THERMODYNAMICS
6. ANALYSIS OF ACIDIC SUBSTANCES BY TITRATION
7. MOLECULAR WEIGHT OF A VOLATILE LIQUID
8. MOLECULAR WEIGHT DETERMINATION BY FPD
9. KINETICS: THE CLOCK REACTION
10. SPECTROMETRIC ANALYSIS OF PHOSPHATE
11. pH, BUFFERS AND THE DISSOCIATION CONSTANT, Ka
12. PAPER CHROMATOGRPAHY

Figure 1.1 shows the structure of the courseware. It consists of a Web laboratory book, student self-assessment tools, a quiz tool, a DISTILS prototype, a faculty authoring tool and several services such as dictionary service and system assessment service. The Web lab book includes the laboratory manual, general chemistry textbook material, and related case studies. Student self-assessment tools are a series of pop-up questions for testing student's understanding of the materials. The quiz tool provides students with a pre-lab test that they must pass before going to the actual laboratory to perform the experiment. Each pre-lab test consists of one numerical question and eight multiple-choice questions. The faculty authoring tool helps faculty member to prepare the pre-lab quiz. The dictionary service provides context-sensitive word definitions to facilitate reading comprehension.

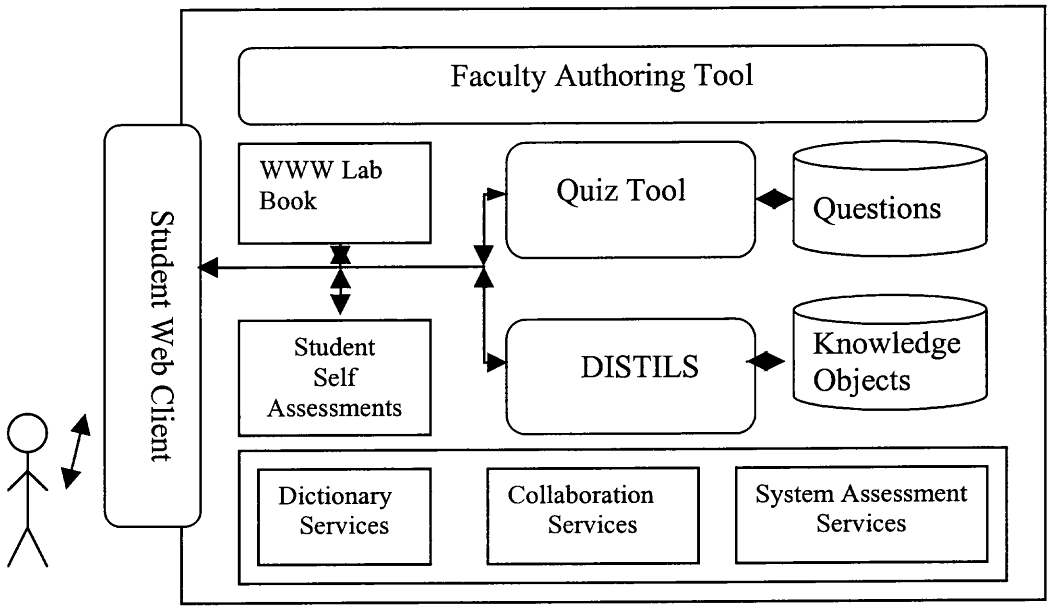


Figure 1.1 The Structure of General Chemistry Laboratory Courseware

As part of this study, DISTILS is incorporated into this complex courseware. The underlying objective is to help the students to help themselves; to encourage and facilitate student-centered learning behaviors and therefore improve students' procedural and analytical skills in the general chemistry laboratory. The adaptive learning system includes: 1) a quiz system for assessment of knowledge, 2) an intelligent tutoring system for those who need help.

1.3 Organization

This dissertation discusses in detail the design of the Web-based adaptive learning paradigm, the design, implementation and evaluation of the prototype--DISTILS. It is organized as follows:

Chapter 1 introduces the importance of and the issues leading to this study, illustrates application background and research objectives, and outlines the dissertation organization.

In Chapter 2, a literature survey is presented. Related research that has had major influence on this study includes is reviewed. The limitations of current related work are pointed out. The Web-based adaptive learning paradigm in this study is discussed. A generic Common Object Request Broker Architecture (CORBA)-driven framework is presented for implementing this paradigm and the advantages of this framework are discussed.

Chapters 3,4,5 and 6 detail the design of components of the prototype system--DISTILS. Knowledge representation, student modeler, intelligent tutoring and adaptive lecture guidance are discussed respectively. Chapter 3 explores an object model for knowledge representation in fundamental chemistry domain. A CORBA-based implementation of the object model is discussed.

Chapter 4 describes the design of the student modeler. The student modeler consists of two components: knowledge practice database and performance predictor. The structure of knowledge practice database and the mechanism of performance predictor are studied.

Chapter 5 describes the design of the Intelligent Tutoring Tool (ITT). The ITT consists of problem space, blackboard and coach delivery components. The structure of problem space, blackboard and the algorithm of coach delivery are presented in detail.

Chapter 6 discusses the Adaptive Lecture Guidance module. The ALG consists of topic extractor, lecture delivery and interface components. The mechanism that brings these components to work cooperatively is presented. This mechanism works together with student modeler in DISTILS to produce navigation guidance sensitive to student's knowledge status.

Chapter 7 discusses the implementation of this prototype system. The evaluation methodology of the prototype system is presented and results found in our evaluation are presented.

Chapter 8 summaries the dissertation and discusses the future research.

CHAPTER 2

LITERATURE SURVEY

This section discusses related studies that have major influence on this study. Recent intelligent tutoring system application results are first presented. The development methodologies of these applications are analyzed and compared. Limitations of the related current methodologies in Web based learning system are also discussed. An object-oriented framework under Common Object Request Broker Architecture (CORBA), a branch of distributed object technology, is then introduced to address these limitations.

2.1 Intelligent Tutoring System (ITS) Applications

Since mid 1980s, falling hardware costs, together with evolving theories of cognitive science and artificial intelligence technologies, sparked a vibrant growth of ITSs. Researchers believed that with the increasing power of modern computer, it is practical to develop computer systems that could serve as private tutors to help students through individualized, human teacher like tutoring sessions. A lot of research has been conducted to develop computer tutors in various areas, like geography (Carbonell 1970), electronic trouble-shooting (Burton and Brown 1982, Lesgold et al 1992), computer game (Goldstein 1982), computer language (Anderson et al 1985a; Anderson and Reiser 1985b; Reiser et al 1992; Sack and Solway 1992; Song and Hahn 1997; Wang 1997), simulation based learning environment (Lester 1996; Schank and Kass 1996), and medical diagnosis (Clancey 1987; Obradovich 1996). Following sections detail some important research in the ITS area.

2.1.1 SCHOLAR, 1970

Carbonell's (1970) SCHOLAR system was the pioneer work in ITS area (Wenger 1987). SCHOLAR taught South American geography. It was capable of holding mix-initiative dialogues with students, responding to their questions by traversing the knowledge network, and asking them questions to convey the content of the knowledge network interactively. When an error happened, the system first attempted to diagnose the

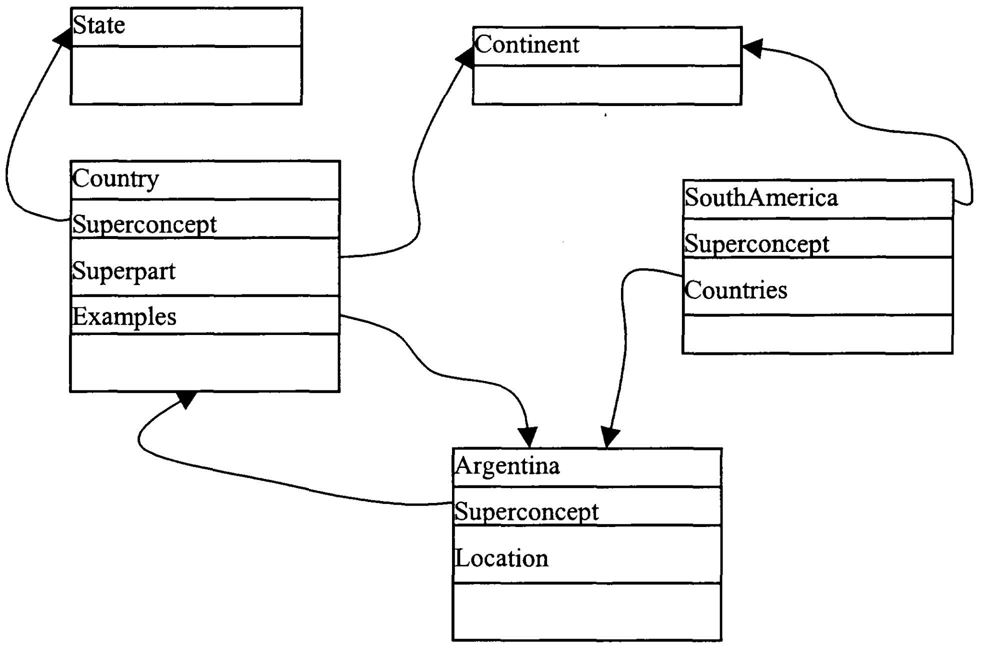


Figure 2.1 A Partial Semantic Graph in SCHOLAR (Carbonell 1970)

student's misconceptions by asking other relevant questions and then presented materials that help the student to see his/her own errors.

The system's knowledge of the geography of South America is represented as a well-defined semantic network of geography objects and concepts. The nodes of this network are units of information-defining words and events in the form of multilevel trees. Figure 2.1 shows a partial knowledge network in SCHOLAR. In the network of Figure 2.1, State, Continent, South America, Country, Argentina are concepts and represented as nodes. The relationships among these concepts are expressed using the arcs. For example, the link between Country and Continent says a country is a part of a Continent, the link between Continent and South America says that South America is a Continent.

2.1.2 WUSOR, 1982

WUSOR developed by Goldstein (1982) was an expert-based coach. WUSOR was designed to foster the student's ability to make proper logical inference from the information given in a computer game called WUMPUS. In WUMPUS game, the player goes through successive caves in a warren where terrible Wumpus is hiding. The player must exercise logical and probabilistic reasoning to decide which neighboring cave to visit next, basing on the signals received so far. The genetic graph was adopted in WUSOR to formalize the syllabus. It served as the basis for tutoring, modeling, and learning. Two components are included in the genetic graph: 1) a rule-based representation of domain knowledge; 2) a learner oriented set of links that captured the evolutionary nature of knowledge, such as generalization versus specialization, analogy, deviation/correction and simplification/refinement.

2.1.3 LISP Tutor and Geometry Tutor, 1985

Anderson and his associates developed LISP tutor and geometry tutor (Anderson et al 1985a; Anderson and Reiser 1985b; Anderson et al 1998). LISP tutor is for novice LISP programmer and geometry tutor is for proofs in high school geometry. Both LISP tutor and geometry tutor embodied a complex cognitive theory - ACT* or more recent version ACT-R. The ACT-R theory is based on a fundamental assumption that there are two types of knowledge-declarative knowledge and procedural knowledge. Declarative knowledge refers to things we know and can usually describe to others. Declarative knowledge is represented in unit of chunks. Procedural knowledge is a type of knowledge that we display in our behavior but we are not conscious of. This knowledge is represented in the form of a production system. Starting from these assumptions, ACT-R explains how the knowledge is deployed and how the knowledge is acquired. Anderson group believed that ACT-R can model how successful students perform various cognitive tasks and therefore could provide a theoretical basis for designing educational software.

Based on ACT-R, both LISP and geometry tutor implemented the following principles: 1) Problem solving is used as tutorial context. In each tutoring session, students first read a statement and then were given several questions to solve. 2) Domain expertise is represented as the ideal model, which is a goal-restricted production system. Knowledge in both elementary LISP and geometry domain was represented as production rules. A set of predefined production rules that are used by experts were defined as the idea model. Students are expected to gradually learn and match this idea model. 3) A model-tracing paradigm is built to infer which rule the student applied by determining which one matches the student's response.

The LISP tutor has produced impressive results. In an experiment, ten students learned from the LISP tutor, ten learned from a human tutor, and ten worked on their own. The human-tutored group took 11.4 hour, the computer-tutored group took 15.0 hours, and the group on their own took 26.5 hours to cover the same material. The three groups performed equally well on the tests of their LISP knowledge. In another test, students using LISP tutor are compared with student learning on their own. Students working with the LISP tutor spent 30 percent less time doing the problems and scored 43 percent better on the final exam.

2.1.4 PROUST, 1992

Sack and Soloway (1992) discussed their experience with PROUST, which has been used by novice Pascal programmers as an on-line debugging aid. Students use a text editor to create their programs and then invoke PROUST to analyze the program, PROUST then writes an output file a description of the errors found. PROUST's knowledge base contains about 37 goals, 55 plans and 70 buggy rules. Given a Pascal program, PROUST is expected to check all the bugs and explain how to fix them. Sack and Soloway studied the impact of PROUST to improve programming performance to find if it helps students find and correct bugs. In their evaluation, students were assigned to ACCESS group and NO ACCESS group. Students in ACCESS group could run PROUST and receive PROUST's analysis in their homework assignment while students in NO ACCESS group could not. Students in ACCESS group performed better on a midterm examination. Students in ACCESS group were also significantly better at fixing program bugs than students in NO ACCESS group.

2.1.5 GIL, 1992

Reiser et al (1992) designed GIL, an intelligent tutoring environment in LISP. The GIL tutor is embedded in a graphical programming environment. Students build a program by connecting together objects representing program constructs into a “graph”. The advantage of GIL is that it provides a more congruent way that novices reason about computer program than text form of programs. Reiser made an empirical comparison of GIL and traditional LISP. In the preliminary test of GIL, eight students used a standard LISP while five students learned LISP using GIL. The time required for two groups to sucessfully solve the assigned problems was examined. The GIL group solved the problems in 2 hours while the text LISP group did in 4.2 hours.

2.1.6 TMT, 1996

Obradovich (1996) investigated the effectiveness of the expert-based Transfusion Medicine Tutor (TMT). TMT was used by medical technology students to learn the identification of alloantibodies in a patient’s blood for finding compatible blood for transfusion. The evaluation results showed that 15 students using TMT went from 0% correct on a pre-test case to 87~93 correct on post-tests, while in the control group, 15 students demonstrated only an improvement rate of 20%.

2.1.7 ELM-ART, 1996

Brusilovsky et al (1996) described their work developing Web-available ITS—ELM-ART. Unlike traditional Web-based textbooks, ELM-ART provides the learner with intelligent navigation support and possibilities to play with examples. ELM-ART uses two adaptive hypermedia techniques – adaptive annotation and adaptive sorting of links.

Adaptive annotation means that the system uses visual cues (icons, fonts, colors) to show the type and educational state of each link. Adaptive sorting is used to present similarity links between cases. ELM-ART knowledge base consists of knowledge about problem solving in LISP that is represented as a network of concepts, plans, and rules. A Common Lisp Hypermedia Server CL-HTTP is used in ELM-ART to handle Web-based interaction.

2.1.8 VNAV&GT-VITA, 1997

At Georgia Tech, Chappell et al (1997) described VNAV and GT-VITA tutor. Both VNAV and GT-VITA tutor were developed for pilot training. VNAV tutor illustrates the operation of the vertical navigation mode using the visualization and animation capabilities of desk-top computers. GT-VITA is a case-based intelligent tutoring system. It includes an intelligent tutoring system and a simulation environment together. In GT-VITA, three types of knowledge—declarative, procedural and operational knowledge—are modeled using operator function model (OFM). An OFM model is essentially a complex hierarchical graph of activities. In the evaluation of VNAV tutor, five line pilots completed six one-hour sessions. The before- and after- test questionnaire showed that participants significantly increased their ability to explain various VNAV operations. At least 80% of the participants correctly performed all tasks in the various categories of VNAV operation.

2.1.9 CyclePad Guru, 1998

Forbus (1998) noticed the attractive capability of developing Internet-based problem solving coaching system and discussed an approach to implement distributed coaching for CyclePad, an environment for learning engineering thermodynamics by design. In this approach, a high performance server—CyclePad Guru was set up to deal with student's request about their associated design, which were sent by e-mail. CyclePad Guru analyzed students' requests and their designs, generate advice, and sent it back to the students.

2.1.10 OWL, 2000

OWL is a Web-based learning environment for General Chemistry instruction (Woolf et al 1999,2000). OWL was first developed as an online quizzing system and then was extended and embedded with guided discovery exercises and intelligent tutoring. Guided discovery exercises allow students to interact with multimedia simulation, which guide them to discover basic laws and concepts such as gas laws. Two intelligent tutors, Stoichiometry Tutor and Lewis Structures, have been developed in OWL. OWL is perhaps hitherto the most extensively tested intelligent learning environment. The OWL system has been used by the full Chemistry courses since the Spring of 1997. In a typical semester over 50,000 Chemistry quizzes are taken, with more than 500 in one day during peak usage periods. The intelligent tutors in OWL produced impressive results. The 30 students who used the Stoichiometry Tutor scored 5-10% better than students in the previous fall.

2.2 Limitations of The Related Work

As presented in above section, many ITS applications had very positive feedback during their evaluations (Anderson et al 1985a; Lesgold et al 1992; Obradovich 1996). However, the development of such system is a very time-consuming and expensive process (Jerinic and Devedzic 2000; Lester 1996; Schank 1994; Wu and Lee 1998). To build an ITS is not easy, it means significant cost of realization, big development team, and large computer resources. It is very attractive to make widely available ITS applications, not only easily accessible from users, but also easily reused by developers. During the development of DISTILS, the following limitations were found in existing ITS research to fulfill this capability.

First, the mechanism for the integration of ITS and the Web has not been well studied. Most existing intelligent learning applications are "on-site" educational systems and tools, which are not widely accessible. While there are other Web-available educational applications, many of them only use the Web as the delivery tool of lecture material. Brusilovsky et al (1996), Forbus (1998), Woolf et al (2000) have done pioneering work in Web-based ITS. However, their methodologies have limitations. CyclePad Guru (Forbus 1998) has an obvious disadvantage that it requires students to have access to email, and its response has a long time delay. ELM-ART (Brusilovsky et al 1996) integrates ITS and Web through Common Gateway Protocol (CGI) and the Common Lisp Hypermedia Server CL-HTTP. However, the CGI method has performance and interoperability problems. The ITSs in OWL were developed as Java applets and were downloaded as a whole and run in the client browser, it was difficult to achieve balanced distribution of computer resource to improve performance.

Second, existing intelligent educational content is hard for other developers to access. Most intelligent educational content today is in the form of massive binders of domain expertise and pedagogical expertise, with little of the work focused on re-usability. Each

ITS has to be built from scratch. Recent research has been addressing this difficulty. Wu and Lee (1998) proposed that systematic development process could effectively improve ITS productivity. Wu and Lee elaborated the ITS design space in terms of the principles of design and highlighted the issues for a systematic approach to ITS, such as a paradigm hierarchy and the need for a description language. However, these issues, as Wu and Lee also noted, are very difficult themselves and somehow are impractical. Authoring tools were also developed to improve the productivity of ITS development (Schank and Kass1996; Jerinic and Devedzic 2000). Although authoring tool makes it easy for developing ITS on a particular domain topic, it does not contribute to the reusability of an ITS. Different systems use different platforms, adopt different architectures and different knowledge organization. It is very difficult to adapt or tailor functions of an existing ITS for purposes other than original ones. New ITS applications have to be developed from scratch, therefore the development cost of is still high.

Another alternative is to develop reusable ITS components that could be easily assembled into new application. However little research has been done so far. Clark (1998) and Roschelle et al (1998) focus on developing reusable educational object to enable plug and play assembly of educational software. Clark's MSE library is a collection of educational animation and simulation tools. It is searchable to find out what may be of interest and to reuse it as a whole. Roschelle et al adopted a component model, OpenDoc, to develop educational components--EduObjects. Although both these two works provided insights in developed intelligent educational components, they had limitations. First, they only focused on the animation and visualization of domain expertise by simulation, with no emphasis on how their animation could adapt to the learner's progress. Second, their objects still cannot be easily reused in environment with different languages and different platforms.

To address the above limitations, two things are critical. The first one is how to model and encapsulate intelligent educational components. The second one is how to provide interoperability of such components in a dynamic and hybrid environment like the Web.

This dissertation made an exploratory effort to design a general object oriented framework for the development of Web-based adaptive learning environment. The objectives are two-fold. The first is to develop and evaluate the effectiveness of an ITS in a Web-based environment; the second is to incorporate the state-of-the-art platform independent distributed object technology to develop reusable, plug and playable Web-based intelligent tutoring components.

In the following sections, a Web-based Adaptive Learning (WAL) paradigm, which integrates intelligent tutoring into a Web-based environment, is first discussed. A general framework for implementing WAL is then introduced.

2.3 Web-based Adaptive Learning (WAL)

2.3.1 Previous Learning Paradigms

The design of a learning environment has always featured and embodied a particular or several learning paradigms. A review of the literature reveals three predominant learning paradigms in the use of computer technologies in education: 1) the objectivism paradigm, 2) the collaborative paradigm, and 3) the constructivism paradigm. Each learning paradigm differs in its underlying cognitive or psychological learning theories.

In the objectivism paradigm, learning is a process of uncritically absorbing objective knowledge of the reality. The teaching is essentially a knowledge communication process. "Knowledge communication is defined as the ability to cause and/or support the acquisition of one's knowledge by someone else, via a restricted set of communication operations" (Wenger 1987). The purpose of teaching is to facilitate the transfer of knowledge from an expert to learners. Historically, objectivism learning paradigm is supported by a behaviorist learning theory where the emphasis was on observable behavior modification. Since mid 1980s, behaviorist theory was replaced with cognitive learning theories that focused on the hidden mental processes in learning under certain situations (Kearsley 1993; Wassom 1997). Many ITSs were developed based on a cognitive theory. For example, LIST-Tutor was designed around ACT-R, a cognitive theory developed by Anderson and his associates.

The collaborative learning paradigm originates from Vygosky's (1962) sociocultural theories of learning, which emphasizes that learning is a social activity which takes place more effectively in the leaner's proximal development zone. It is claimed that knowledge can be easily converged and formulated through peer communication or communication between teachers and learners (Roschelle 1996). The more it is shared and discussed, the more it can be learned and the more consistent it is. Communication, listening, and participation are key factors in improving learning efficiency. A collaborative learning environment always tries to facilitate conversations, discussions and debates among participants to maximize the sharing of information and knowledge among learners (Fuji 1996; Koschmann 1996; Li 1998; Wang 1997). Numerous studies have demonstrated the effectiveness of collaborative learning.

Roschelle (1996) showed that a group of students is more likely to attain the same level of understanding and mastery through collaborative learning efforts, while such results are not achievable through individual learning efforts. Li (1998) showed also that students were more active at answering questions from their peers and had better performance in their final examination when they are taught in a collaborative environment.

The constructivist paradigm had its origins in the work of the developmental psychologist Piaget (1977), who advocated the idea that each individual assimilates and accommodates prior knowledge and therefore constructs his or her own interpretation of an objective world. In contrast to objectivist paradigm, the constructivist paradigm views the learning process as an active, goal-oriented, and constructive process instead of just a knowledge transfer process. The knowledge is created by learners, rather than transmitted to learners. Individuals perform better in learning when they are forced to discover things themselves rather than when they are instructed. The constructivist view of learning has inspired the development of a number of instructional methods, eg. "Learning by designing" (Lester 1996), "Simulation-based learning by doing", "Learning by exploring" (Schank 1994), and “learner-centered learning” (Norman and Spohrer 1996), all of them dedicated to the proposition that individuals perform better in learning when they are under circumstances of personal inquiry and discovery. For example, Lester et al (1996) argued that in their formative evaluation of the Design-A-Plant, most of students' time was spent in making designing decisions about features of the design that were most critical. In Broadcast News (Schank1996), students are assigned a story to work on. A rough draft of the story is produced for the student to take as a starting point. The student's main task is to edit the text and video to match the critique from subject matter experts.

2.3.2 Web-based Adaptive Learning

Web-based Adaptive Learning (WAL) model integrates intelligent tutoring and constructive learning paradigm. The objective of WAL is to help the students to help themselves and to encourage and facilitate student-centered learning behaviors in the Web environment. In this paradigm, intelligent educational objects are incorporated to facilitate students' constructive learning through navigating, searching and practicing by addressing certain challenges intrinsic in the Web learning environment. Its main features are summarized in the following sections:

Learning by Navigating vs. Adaptive Navigation Guidance

The hypermedia format of the Web can provide students an environment to search knowledge in a way that suits their logical needs. Students can browse the content at will: they can decide to follow a link, to re-do a section, to totally skip a chapter or to complete an exercise. Students are active and creative participants in navigating and exploring the knowledge space. It has been found that this feature is most suitable for knowledgeable users who know what they are seeking (Gray 1993; Ashani 1998). However, especially when students are novices in the subject matter, the non-linear feature of the hypermedia education can contribute to learning difficulty; students may be easily lost, memory overloaded, or may miss or be unable to find important contents in a large non-linear hyper-space. In this study, adaptive navigation guidance is introduced to encourage students and make them feel successful during their learning exploration. It monitors the progress of student knowledge level and indicates where they are and what they are missing or if they are not well prepared. In this way, students easily get what they need and therefore can take full advantage of the Web-based learning environment.

Constructive Learning by Doing Using an On-line Intelligent Tutoring

Students form their own mental maps of domain subjects by navigating Web-based knowledge space with self-adaptive guidance. Their mental maps need to be verified and optimized so that they can effectively solve practical problems. "Learning by doing" is often the best way for students to test and validate their knowledge. Students could learn how it works when applying their own knowledge, during which incorrect or correct understandings could be corrected or strengthened, respectively. Entire mental maps therefore could be gradually optimized. However, there are dangers inherent in this sort of learning by doing, especially in Web-based environment where timely help cannot be guaranteed. Floundering during problem solving can often lead to confusion and frustration. An on-line intelligent tutoring tool could significantly alleviate these problems and help students acquire the necessary procedural skills.

2.4 A CORBA-driven Object Oriented WAL Development Framework

There are two essential issues to be addressed for the development of reusable intelligent tutoring components in a heterogeneous environment like Web. The first issue is how to encapsulate knowledge components, and the second issue is interoperability. Existing ITS research, however, does not address these two issues well. In this study, object oriented knowledge representation techniques, together with Common Object Request Broker

Architecture (CORBA), an industry standard toward developing distributed and interoperable software components, are used to integrate ITS components in Web based environment.

In the following sections, first a review of knowledge representation techniques has been provided. Distributed object technology--CORBA is then briefly introduced. Finally, a framework is designed using CORBA as the infrastructure to facilitate reusable WAL application development.

2.4.1 Knowledge Representation

Knowledge representation has been a central problem in most ITS implementations. The basic problem of knowledge representation is to provide sufficient presentation notation with which to represent and process knowledge (Wang 1997). Three important techniques have been found in knowledge representation area: semantic networks, production rules, and object oriented methods (Carbonell 1970; Anderson et al 1985a; Adeli and Hung 1990; Lee and O'Keefe 1996; Gorti et al 1998).

Semantic network has its origin in natural language processing (Bench-capon 1990). In linguistics, the study of semantics attempt to describe the meanings of words, and semantic networks attempt to give this description by relating the symbols in a network. There are two important components in a semantic network: nodes and arcs. The nodes represent concepts denoted by the words, and the arcs represent relationships between these concepts. Figure 2.1 shows a simple example of a semantic network.

The use of semantic network in ITS traced back to the work done by Carbonell (1970). In SCHOLAR, semantic networks were used to store and access information and served as the basis to convey knowledge to students. Webb's feature networks in AICAL is a kind of enhanced semantic network (Webb 1988). OFM (Chappell et al 1997; Chu et al 1995) in GT-VITA is a much complex semantic network. In OFM, the nodes represent operator activities that need to be learned, and the arcs define events critical for control. Although SCHOLAR, OFM, and AICAL demonstrated that semantic networks could effectively support knowledge representation and processing in ITS, these networks do present some fundamental limitations; there is a certain lack of structure, which leads to realistically sized networks becoming extremely complicated. Also in semantic networks, it is difficult to represent procedural knowledge (Wenger 1987).

The production system technique has been the major knowledge representation technique (Anderson and Lebiere 1998). A production system consists of a set of rules in which each rule represents a unit of skill. A rule has two components. The first component is a list of conditions and the second component is a list of actions which may be appropriately performed when the conditions are satisfied. Given a set of initial data structures, a production system operates as follows. That rule whose condition is true of current data is fired, that is the actions are taken. The results modify the current data structures. This leads to another rule being fired, leading to further modification. The entire process halts either when no condition is true or when an action containing a stop operation occurs.

The following is a simple example of production rule.

```
IF (?x bird true)
THEN (?x canFly true)
```

This rule says that if it is a bird, it then can fly.

The major advantage claimed for production system is that it provides a single uniform method of representation, which is relatively easy to understand for the non-computer specialists. The *IF...Then* format is intuitively straightforward, and the use of these rules is not difficult to grasp. Due to such advantages, production systems have been a dominant technique in knowledge representation area and have been used by majority of intelligent tutoring applications. Most intelligent system applications adopted production system. So do ITSs (Goldstein 1982; Anderson et al 1985a; Anderson and Reiser 1985b; Brusilovsky et al 1996; Forbus 1998; Wong and Quek 1998).

An important disadvantage of production system is its maintainability. While it seems easy to realize incremental development by adding new rules, a lot of issues are concerned, such as what other rules there are, and how they will interact with the new one once when these rules are not truly declarative. Large production systems tend to become hard to update. The lack of structure in production systems eventually starts to become a drawback. It has been suggested that productions rules are inadequate for describing domain objects and as well as for describing static relationships among objects (Lee and O'Keefe, 1996).

Object oriented knowledge representation has its origin in object-oriented programming language and object oriented software engineering. The increasing popularity of object oriented programming language and wide success in the object oriented software engineering practice (Jocobson 1993; Booch 1994; Chien and Xue 1997) leads to the introduction of object oriented technology into knowledge representation area. Object oriented programming language provides a more natural way for developing software components. The basic mechanisms of object oriented programming language are objects, messages, methods, classes, subclasses, and inheritance and polymorphism. Objects are discrete software modules that contain both data and instructions that operate the data, thus objects have the ability to act. Action occurs when an object receives a message. Methods that reside in an object determine how the object acts when it receives a message. Packaging an object's data within the protection of its methods is called encapsulation. A class is a description of a set of nearly identical objects. Each object is an instance of its class. Classes are organized into class hierarchies. Subclasses inherit state and behavior from their superclass, and usually have some additional attributes and methods of their own. Inheritance provides a natural way for structuring software modules, which increases their modularity and reusability.

The essence of problem solving within the object-oriented paradigm is to identify the real-world objects relevant to the problem, the attributes of those objects, and the processing operations in which they participate. Many practices have shown that by encapsulating highly related data and methods into objects and organizing objects into a hierarchical class structure, object technology make it possible for us to manage large systems, to change them and to reuse parts of old systems in new systems (Booch 1994; Jacobson 1993). The object-oriented mechanism embedded into the programming languages like C++ enables the programmer to reuse code efficiently (Neyer and Wu 1990).

Object oriented knowledge representation has become popular since the early 1990s. Having seen the advantages of object technology in software engineering, researchers are interested to see if these advantages could be taken in the area of knowledge representation. In an object oriented knowledge representation scheme, knowledge elements, ranging from primitives to complex elements, are represented as knowledge classes. A knowledge class is an integration of both declarative and procedural knowledge on a particular topic. Declarative knowledge is represented in attribute, and procedural knowledge is implemented as methods. Among knowledge classes, there are general relationships such as generalization/specification, aggregation, or domain related relationships, such as supervision between teachers and students, neutralization between acids and bases. Knowledge classes, together with the relationships, usually form a hierarchical structure, which facilitates knowledge organization and improves reusability. A well-defined knowledge class could be easily inherited and extended in such hierarchical structure.

Recently, object oriented knowledge representation techniques have been widely studied in different domains (Akagi 1990; Kurumbalapitiya and Ratnajcevan 1993; Lefancois and Montreuil 1994; Raphael and Kumar 1997; Tang 1997; Gorti et al 1998; Karacal and Mize 1998; Ming and Yang 1998; Hakman and Groth 1999). Chien (1997) developed an object oriented knowledge representation scheme into the task planner for mobile navigation. Raphael and Kumar (1997) proposed object oriented knowledge representation scheme to deal with past cases in the development of case based design systems. Gorti et al (1998) investigated an object-oriented knowledge representation for product and design processes, and found that object-oriented approaches have enabled a natural decomposition and hierarchical structuring of design product knowledge. Emelyanov and Iassinovski (1997) adopted an object-oriented approach for discrete manufacturing systems simulation and concluded that the use of the object-oriented approach increases the clarity and ease of manufacturing systems description for simulation. Tang (1997) applied object technology in building environmental modeling and observed that the development of object-based knowledge representation mechanism enables the compatibility and extensibility. Hakman and Groth (1999) concluded that both the object-oriented and distributed objects methodologies were more feasible and suitable for biomedical system modeling than that was available before such as continuous modeling and others. Vaishnavi and Buchanan (1997) describe smart object paradigm as suitable knowledge representation paradigm for the modeling and design of operations support systems. A smart object is an encapsulation of task knowledge, control knowledge, data, and procedures, which can be used to model complex operations environment. The object oriented approach has also been attempted in ITS. Wang (1997) presented an object-oriented approach for knowledge representation scheme in a collaborative learning environment—LearnOOP. In LearnOOP, Telos, an object oriented knowledge representation language, was successfully applied to model the knowledge elements like topics and algorithms.

2.4.2 Common Object Request Broker Architecture (CORBA)

CORBA is a middleware specification developed by Object Management Group (OMG), a consortium of over a hundred of companies to facilitate distributed object computing (OMG 2000). It has two major objectives: 1) to make it easier to implement new application in distributed, hybrid-environment, such as different hosts, different languages. CORBA makes networking programming much easier by allowing one to create distributed applications that interact as though they were implemented in a single programming language on one computer; 2) to encourage the writing of open, reusable applications to improve productivity of software engineering (IONA 1998). To do this, CORBA brings the advantages of object-oriented techniques to a distributed environment. It allows one to design a distributed application as a set of cooperating objects and to reuse existing objects in new applications.

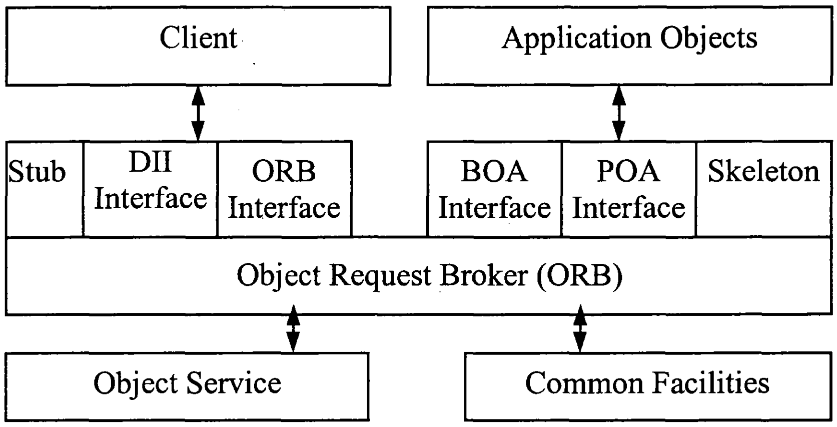


Figure. 2.2 CORBA Framework

As shown in Figure 2.2, four major parts are defined in CORBA: the Object Request Broker (ORB), Object Services, Common Facilities, and Application Objects (Seigel 1996; Pope 1997; Hoque 1998). Object Services defines the system-level frameworks necessary for any application to be constructed in a reasonably high level, including object life cycle, naming, event notification, persistence, transactions, and concurrency. Common Facilities are application-oriented objects providing high level functionality that defines general capabilities required by many applications, including accessing databases, printing files, document management, and e-mail. The ORB sits at the architecture's heart. It provides the basic object interaction capabilities necessary for communication among any of the components. It lets objects transparently make requests to and receive responses from other objects locally or remotely. Application Objects represent the actual software being developed for solving domain-specific problems.

CORBA relies on a protocol called the Internet Inter-ORB Protocol (IIOP) for remoting objects (Baker 1997). Everything in the CORBA architecture depends on the ORB. The ORB acts as a central object bus over which each CORBA object interacts transparently with other CORBA objects located either locally or remotely. Each CORBA server object has an interface and exposes a set of methods. To request a service, a CORBA client acquires an object reference to a CORBA server object. The client can now make method calls on the object reference as if the CORBA server object resided in the client's address space. The ORB is responsible for finding a CORBA object's implementation, preparing it to receive requests, communicate requests to it and carry the reply back to the client. A CORBA object interacts with the ORB either through the ORB interface or through an Object Adapter - either a Basic Object Adapter (BOA) or a Portable Object Adapter (POA).

For developing CORBA-compliant application objects, developers need to specify an interface for each object by using Interface Definition Language (IDL). IDL is one of the founding principles of CORBA. It is CORBA's object contract language. An Object's IDL defines it boundaries in terms of its contractual interfaces with potential objects. The IDL interface files describes the data types and methods or operations that a server provides for an implementation of a given object. Since the OMG does specify mappings from CORBA IDL to various programming languages, including C, C++, SmallTalk, and Java, CORBA Objects written in IDL are portable across languages, tools, operating systems, and networks. CORBA provides operating system and language independent interfaces to all the services and objects residing on the ORB. Therefore, client and server objects written in different languages and running on different platforms can inter-operate through ORB and its associated service provided by ORB vendors.

CORBA has been successfully applied in the development of applications in many domains: finance, healthcare, manufacturing, education and commerce (Gennari et al 1998; Narayanan and Malu 1998; Develzic and Radovic 1999; Hoffiner et al 2000; Rezayat 2000). Narayanan (1998) applied CORBA in developing Web-based airbase logistics system simulation. Rezayat (2000) recommend combining CORBA with the Web standards to create the object web to deal with integrated product and process development in distributed design and manufacturing environment. Harvard University successfully deployed CORBA in the development of course-online system (OMG 2000). Gennari et al (1998) developed an architecture for a CORBA implementation of a library of platform independent, sharable problem-solving methods and knowledge bases. In their approach, CORBA provides a useful infrastructure that helps components in a reuse library more accessible.

2.4.3 A CORBA-driven WAL Development Framework

A generic CORBA-driven object oriented framework was designed for developing Web-based adaptive learning components. The framework is shown in Figure 2.3. It is based on a layered structure and essentially consists of the following six components:

- **User Interface Components**: These are components that collectively define the web-based user interface and developed by using HTML, and Java.

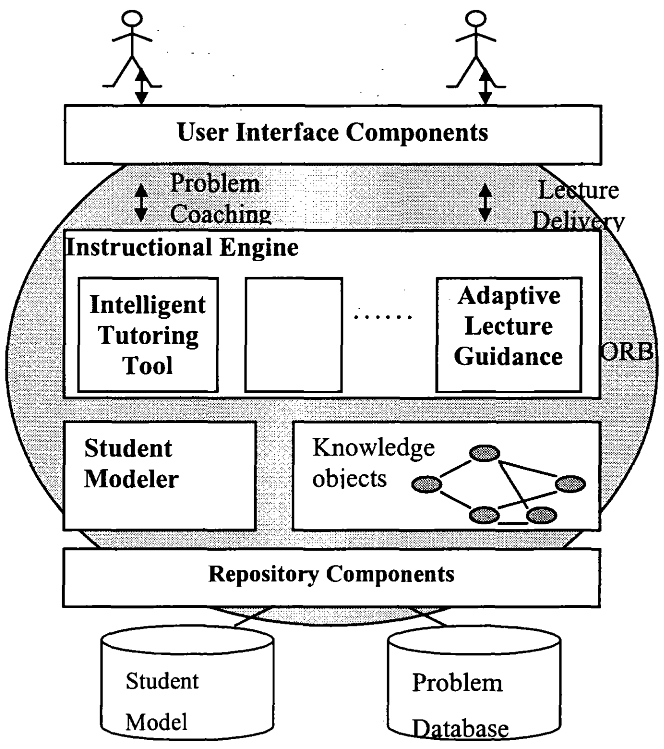


Figure 2.3 A CORBA-driven Framework for WAL

- **Repository Components**: These components provide facilities to provide connectivity to data sources and load and store objects across distributed environments.
- **Knowledge Objects**: Knowledge objects model domain expertise. Each knowledge object owns expertise on a small topic. In DISTILS, a knowledge object consists of capability of problem solving, explanation, and interoperability. Small knowledge objects could interact together and represent expertise on a large topic.
- **Student Modeler**: Student modeler traces student's knowledge practice and predicts student's performance on a particular knowledge in a particular problem context.
- **Instructional Engine**: Instructional engine is the core educational component in this framework. It is responsible for generating context sensitive tutoring behaviors. It

could consist of a variety of tutoring tools in terms of requirements of particular applications. In the DISTILS prototype, it consists of two major components: **intelligent tutoring tool (ITT),** and **adaptive lecture guidance (ALG).**

Intelligent Tutoring Tool (ITT): ITT acts as a problem solving coach. Students who have difficulty in passing pre-lab quiz are expected to get on-line problem solving coaching from ITT. It is composed of three modules: Current Problem Space, Blackboard and Coach Delivery. Current problem space stores knowledge objects, which are initialized in terms of problem statements. These knowledge objects are expected to solve the questions collaboratively and to generate and put solution plan and activities on Blackboard. Coach Delivery reads the solution plan and activities from blackboard and schedules tutoring topics in terms of student knowledge status.

Adaptive Lecture Guidance (ALG): ALG acts as student curriculum advisor. Usually it works at passive role. In DISTILS, **ALG** passively responds to student request for help if students do not know where they are and what is the appropriate next learning topic. The student modeler is then activated to generate student knowledge status tree; and the topic extractor finds the prerequisite relationships among student unknown topics; finally the lecture delivery determines the selection of the next topic and illustrates to the student where s/he is and how to get to the topic suggested.

- **ORB**: ORB refers to CORBA middleware that mediates the transfer of messages from a program to an object located on a remote network hosts. In this study, Java-

ORB serves as the communication backbone hiding the underlying complexity of networking communication from the programming.

This CORBA-based framework provides a more flexible mechanism than existing methods for the development of WAL. First, it eliminates the platform restrictions for existing applications (Gennari et al 1998). Second, intelligent educational applications usually need huge computer resources (Forbus 1998). With the support of this framework, applications are more scaleable. Intelligent educational components could be easily deployed over distributed computer resources and the system load will be more balanced.

Another advantage of the CORBA-driven framework is that it makes possible for the development of reusable, plug and playable intelligent educational components. Well-developed educational components can be reused as-is in new applications given that their interfaces are well defined using CORBA IDL, and new components can be built by making incremental modifications to existing ones. As shown in Figure 2.4, instruction engine consists of two components: intelligent tutoring tool and adaptive lecture guidance. New components could also be developed and easily assembled into the system. One could also designe new knowledge objects and a student modeler and easily assemble them with existing systems given they obey each other's interface specifications.

CHAPTER 3

KNOWLEDGE OBJECTS: AN OBJECT KNOWLEDGE MODEL

3.1 The Knowledge Features in General Chemistry Laboratory

To be able to represent Chemistry in terms of interacting objects, it is fundamental to define its characteristics. Chemistry deals with all the substances that make up our environment. It also deals with the changes that take place in these substances - changes that make the difference between a cold and lifeless planet and one with life and growth (O'Connor 1977). Particularly, fundamental chemistry deals with common properties of daily substances, regularities among these properties, and regularities about reactions among these substances.

Knowledge of general chemistry could be considered substance-oriented. It could be viewed from two different perspectives: static and dynamic. 1) From a static perspective, each substance has its own properties identifying itself, for example water (H_2O) is described by the color, form, composition, boiling temperature, and other properties. Different substances have different compositions and properties. For example, the water molecule is composed of two hydrogen atoms and an oxygen atom, while the hydrogen chloride molecule is composed of a hydrogen atom and a chloride atom. Substances sharing important similar properties are classified into the same group, for example salts are classified into electrolytes as they all dissolve in water to give solutions that conduct electricity. 2) From a dynamic perspective, changes take place in these substances. Some changes are internal in a substance, for example, the density of water changes with its temperature, and the volume of gas changes accordingly when pressure changes. Some changes are results of the reactions among different substances under certain conditions, for example, hydrogen and oxygen react together producing water and releasing energy. Regularities are observed during these changes. For example, the density of water is found to change in terms of temperature according to certain formula, and in chemical reactions, mass and atoms are found to be conserved when the chemical bonding changes.

In conclusion, knowledge in general chemistry can be represented in two major groups. One is about substances, like water, vinegar, their properties, like form, density, and their compositions. The other is chemical interrelationship among these substances, in other words, how substances react together. Students need to learn such knowledge to develop their operational skills of how to take advantage of them to identify unknown substances or measure properties of unknown substances. Further, they need to learn and develop skills to use their knowledge in a quantitative way to calculate an amount produced, or rates of reactions, or amounts of energy consumed or released, etc.

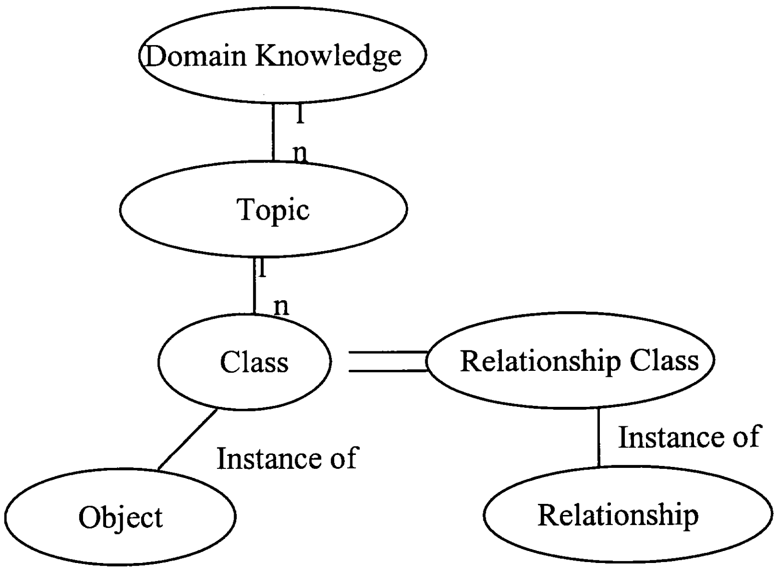


Figure 3.1 Organization of Domain Knowledge

3.2 An Object Model

Using an object model, we formalize the fundamental general chemistry domain in terms of objects and relationships. The organization of domain knowledge is illustrated in Figure 3.1. The domain knowledge consists of multiple topics. A topic describes a tutoring objective. It consists of related knowledge classes and relationship classes. A knowledge or relationship class describes a single knowledge element of domain knowledge and is elaborated in the next section. Knowledge objects and relationships are instances of knowledge class and relationship class.

3.2.1 Definition of Objects

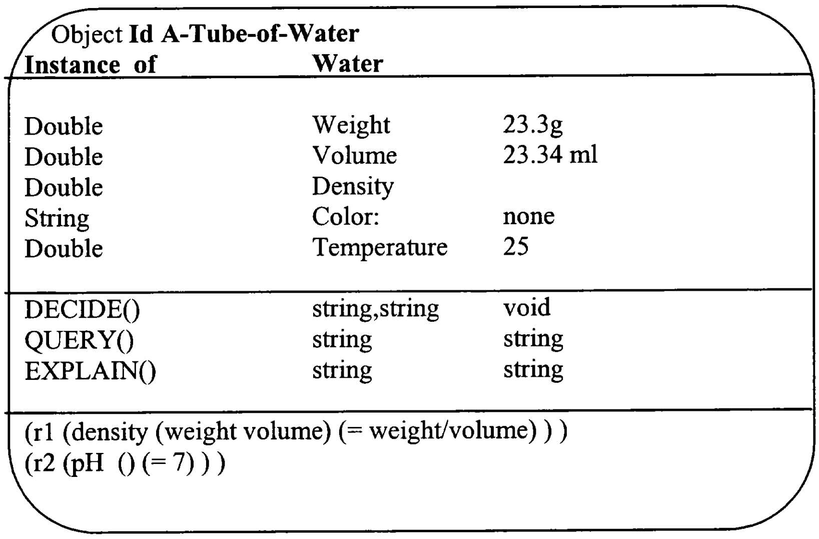
An object, **Co**, is defined as a unique, identified entity in the following form:

Co = (Id, A, R, M, K)

- **Id** is the unique identifier of an object (**Co**).
- **A** is set of attributes. Each attribute is represented in terms of three-tuples, **(t, a, v)**. **a** is the name of the attribute of **Co** and is represented by a symbol which is unique in **A**. **t** is the type of the attribute and **v** is the value.
- **R** is a tuple, **(drl, dr2, .., drn)**. **drl, dr2**, …, and **drn** are the name of the dynamic relationships the object is involved.
- **M** is a set of tuples, ($m, tp_1, tp_2, \ldots, tp_n$, tr). Each tuple is a method signature. The symbol m represents a method name; methods define operations on objects. The symbols, $\mathbf{tp_1}$, $\mathbf{tp_2}$, **…**, $\mathbf{tp_n}$, specify the argument type and **tr** specifies the returned value type.

- **K** is a set of knowledge units owned by **Co.** A knowledge could be either factual or procedural. Factual knowledge is used to determine the value of attributes; procedural knowledge defines complex fuctional relationships among attributes. Each knowledge within the set is defined by (kname, descrip), where kname is a unique identifier for the knowledge, and descrip is its description. Each description often takes the form of symbolic expression, **(ad dd exp)**. **ad** is the target attribute. **dn** is the dependent attributes, and exp is the expression of the knowledge. **exp** takes the form of (= **formula**). **formula** defines the value relationship among target attribute, **ad**, and dependent attributes, **dd**.

Objects can model substances, energy and particles in general chemistry. Objects use the encapsulation method to contain a collection of related attributes and methods. The attributes represent properties identifying substances or energy or particles and their



Object **Id A-Tube-of-Water**

Instance of	**Water**	
Double	Weight	23.3g
Double	Volume	23.34 ml
Double	Density	
String	Color:	none
Double	Temperature	25
DECIDE()	string,string	void
QUERY()	string	string
EXPLAIN()	string	string

(r1 (density (weight volume) (= weight/volume)))
(r2 (pH () (= 7)))

Figure 3.2 A Simple Object: a tube of water

states. The methods are the procedures for manipulating attributes.

For example, let us consider **a tube of water** as an object (Figure 3.2). Water has its properties like weight, volume, density, etc., which are stored as attribute values. It also has methods like DECIDE, QUERY, and EXPLAIN. The DECIDE method in water object could be used to assign a value to any property. The QUERY method could be used to query the value of any property, if the value is not known, it will automatically try to solve according to available information. EXPLAIN method could be used to ask water object to explain its answer. The water object also possesses two knowledge units. One is how to measure the density given the weight and the volume. The other is a factual knowledge, which defines the value of the pH attribute as 7. The example is shown in Figure 4.2 and is formally specified as follows:

```
(A-Tube-of-Water,
        (
                (double weight 23.3)
                (double volume 23.34)
                (double density)
                 (string color none)
                (double temperature 25)
        )
        (
        )
        (
                (DECIDE() string, string, void)
                (QUERY(), string, string)
                (EXPLAIN(), string, tutorContext)
        )
        (
                (r1 (density (weight volume) (= weight/volume) ) )
                (r2 (pH  () (= 7) ) )
        )
).
```

3.2.2 Relationships

The object model explicitly represents relationships among objects. A generic relationship is defined as follows:

R=(rid, RO, A, M)

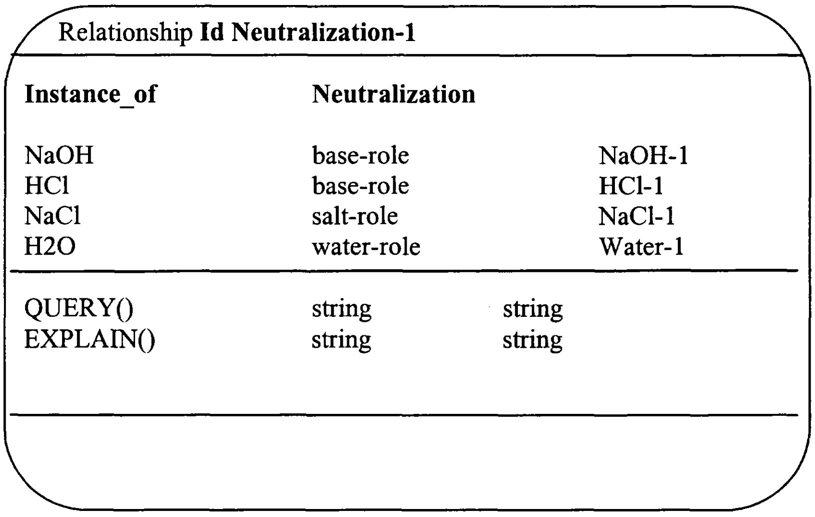
- **rid** is the unique identifier of the relationship.
- **RO** is a set of three-tuples, (**t, ro, v**). Each tuple of **RO** is called a role in the relationship. **ro** is the name of the role and **v** is the value of role and **t** is the type of **v**. types of values can be such as acid role and base role for a neutralization relationship. There must be at least two objects to define the roles in a relationship.
- **A** is set of attributes. Each attribute is represented in terms of three-tuples, (**t, a, v**). **a** is the name of the attribute of **ro** and is represented by a symbol which is unique in **A**. **t** is the type of the attribute and **v** is the value.
- **M** is a set of tuples, (m,tp_1,tp_2,…,tp_n, tr). Each tuple is a method signature. The symbol m represents a method name; methods define operations on objects. The symbols, **tp_1,tp_2,…,tp_n**, specify the argument type and **tr** specifies the returned value type.

Figure 3.3 shows a relationship of neutralization among a NaOH object, NaOH-1, and a HCl object, HCl-1. It is also formally specified as:

```
(neutralization-1
    (
            (NaOH, base-role,  NaOH-1)
            (HCl, acid-role, HCl-1)
            (NaCl, salt-role,  NaCl-1)
            (H2O, water-role, Water-1)
    )
    ()
    (
            (QUERY(), string, string)
            (EXPLAIN(), string, tutorContext)
```

)

).



Relationship **Id Neutralization-1**

Instance_of	**Neutralization**	
NaOH	base-role	NaOH-1
HCl	base-role	HCl-1
NaCl	salt-role	NaCl-1
H2O	water-role	Water-1

QUERY()	string	string
EXPLAIN()	string	string

Figure 3.3 A Relationship: Neutralization of NaOH-1 with HCl-1

3.2.3 Classes

A Class is a template shared by similar objects. It is defined in the similar form as an object:

Cc = (Idc, Ac, Rc, Mc, Kc)

- **Idc** is the unique identifier of a class **(Cc).**
- **Ac** is set of attributes. Each attribute is represented in terms of bipartite, **(t, a)**. **a** is the name of the attribute of **Cc** and is represented by a symbol which is unique in **Ac**. **t** is the type of the attribute, such as string, float, double and integer.
- **Rc** is a tuple, **(drl, dr2, .., drn)**. **drl, dr2**, ..., and **drn** are the name of the relationships the class is involved.

- **Mc** is a set of tuples, $(m,tp_1,tp_2,\ldots,tp_n, tr)$. Each tuple is a method signature. The symbol m represents a method name; methods define operations on classes. The symbols, $\mathbf{tp_1}$, $\mathbf{tp_2}$,..., $\mathbf{tp_n}$, specify the argument type and **tr** specifies the returned value type.
- **Kc** is a set of knowledge units owned by **Cc.** A knowledge could be either factual or procedural. Factual knowledge is used to determine the value of attributes; procedural knowledge defines complex functional relationships among attributes. Each knowledge within the set is defined by **(kname, descrip)**, where **kname** is a unique identifier for the knowledge, and **descrip** is its description. Each description often takes the form of symbolic expression, **(ad dd exp)**. **ad** is the target attribute. **dn** is the dependent attributes, and exp is the expression of the knowledge. **exp** takes the form of **(= formula)**. **formula** defines the value relationship among target attribute, **ad**, and dependent attributes, **dd**.

Similarly, a relationship class is a template shared by similar relationships. It is defined in a similar form as a relationship:

Rc=(ridc, ROc, Ac, Mc)

- **ridc** is the unique identifier of the relationship class.
- **ROc** is a set of bipartite, **(t, ro)**. Each tuple of **ROc** is called a role in the relationship. **ro** is the name of the role and **t** is the type of **ro**. There must be at least two classes when defining the roles of a relationship.
- **A** is set of attributes. Each attribute is represented in terms of pair-tuples, **(t, a)**. **a** is the name of the attribute of **ro** and is represented by a symbol which is unique in **A**. **t** is the type of the attribute.

- **M** is a set of tuples, ($m,tp_1,tp_2,\ldots,tp_n$, tr). Each tuple is method signature. The symbol m represents a method name; methods define operations on objects. The symbols, **$tp_1,tp_2,\ldots,tp_n$**, specify the argument type and **tr** specifies the returned value type.

Class is an abstraction mechanism that makes common properties and semantics. Classes define methods and attributes that can be inherited by **subclasses** or objects. An object is classified as an instance of a class if the object inherits all attributes, relationships, methods and knowledge of the class. Generalization and specialization are also defined in terms of the class abstractions. Special cases of a class are commonly known as subclasses of that class; the more general case, in turn, is known as the **superclass** of its special cases. Generalization/Specification is a special case of relationship class, and it is represented as follows:

```
G/S = ( G/S  (
                   (class  "superclass role" )
                   (class  "subclass role" )
             )
             (
             )
             (
             )
      )
```

In general chemistry, generalization happens when substances are classified or regularities are found. For example, gases, liquids, and solids have different structure or features, but they do have some common properties and same regularities over these properties, like volume, weight, and density, and a common relationship regularity between density, volume and weight. Therefore, substance could be considered as a generalization of liquid, solid, and gas, as shown in Figure 3.4.

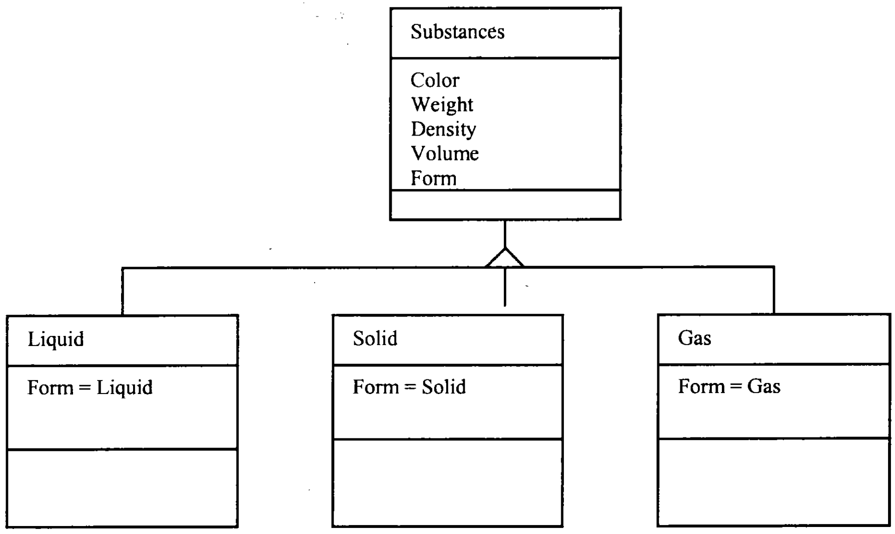


Figure 4.4 An Example of Class Inheritance

Composition defines the structural relationships among classes or objects. It is another special case of the relationship class. It is represented as follows:

```
Comp = ( Comp      (
                         (class "container role")
                         (class "part role")
                   )
                   (
                         (double nContainer)
                         (double nPart)
                   )
                   (
                   )
       )
```

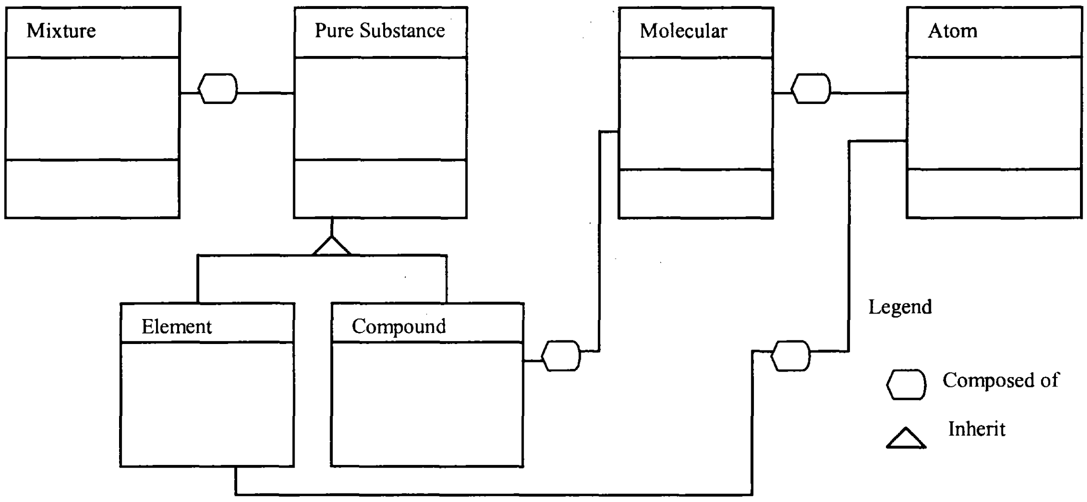


Figure 3.5 An Example of Composition Relationship

Figure 3.5 shows some composition relationships in the general chemistry domain. A mixture is two or more substances combined in varying proportions - each retaining its own specific properties. The components of a mixture can be separated by physical means, without the making and breaking of chemical bonds, such as air, table salt thoroughly dissolved in water, milk and wood. A pure substance refers to a substance with a constant composition. It can be classified as either an element or as a compound. Examples: Table salt (sodium chloride, NaCl), water (H_2O), iron (Fe), copper (Cu), and oxygen (O_2). An element is a substance that cannot be separated into two or more substances by ordinary chemical (or physical) means. Elements are composed of only one kind of atom. Examples: Iron (Fe), copper (Cu), and oxygen (O_2). A compound is a substance that contains two or more elements, in definite proportion by weight. The composition of a pure compound will be invariant, regardless of the method of preparation. Compounds are composed of more than one kind of atoms. The term molecule is used for the smallest unit of a compound that still retains all of the properties of the compound. Examples: Table salt (sodium chloride, NaCl) and water (H_2O).

3.2.4 Topic

Classes and objects that are closely related and generally learned as a group are modeled together as a topic. A topic may correspond to a part, a chapter, a section, or a subsection of a textbook, whose content consists of a subset of the subject materials. A natural criterion for forming topics is to group classes that have the same tutoring goal. A generic topic is defined as follows:

T=(tid, TC, TR)

- **tid** is the unique identifier of the topic.
- **TC** is a set of classes. Each element of **TC** is a domain concept that is modeled as object classes.
- **TR** is a set of relationship classes. Each element of **TR** is a regularity that is modeled as relationship classes.

For example, Matter is one of the tutoring topics in General Chemistry laboratory. It consists of concepts such as SUBSTANCE, MIXTURE, SOLID, GAS, LIQUID, and relationships such as Contain.

3.3 Prototype General Chemistry Object Model Description

The knowledge classes and relationships defined in the prototype system under the defined object model are described. Specifically, we focus on the design of the object model in three laboratory experiments: Density, Acid Titration, and pH and Buffers.

3.3.1 Classes

Table 3.1 shows 14 knowledge classes that are defined for the above three laboratories.

Table 3.1 A List of Knowledge Classes in The Prototype System

Knowledge Class	Applied Laboratory
DomainAgent	Density, Acid Titration, pH Buffer
Substance	Density
Atom	Acid Titration, pH Buffer
Molecular	Acid Titration, pH Buffer
Solid	Density
Water	Density
Acid	Acid Titration, pH Buffer
Base	Acid Titration, pH Buffer
NaOH	Acid Titration, pH Buffer
KHP	Acid Titration, pH Buffer
Vinegar	Acid Titration, pH Buffer
HCl	Acid Titration, pH Buffer
FormicAcid (Acetic Acid)	Acid Titration, pH Buffer
Nicotinic	Acid Titration, pH Buffer

The DomainAgent class is an abstract base class describing common properties of all knowledge classes in a knowledge base. Since DomainAgent is the most generic class in the prototype object model, it is placed in the very root of the class diagrams. The DomainAgent class is defined as follows:

```
        (DomainAgent(
                        (string name)
```

```
                    (string definition)
                    (string extension)
                    )
                    (
                    )
                    (
                            (DECIDE() string, string, void)
                            (QUERY(), string, string)
                            (EXPLAIN(), string, tutorContext)
                            (RegistRel(), Relationship, void)

                    )
                    (
                    )
            )
```

The DomainAgent has three attributes: name, definition, and extension that provide general information about a class. There are four methods that are also defined in DomainAgent and explained as follows:

- Query: Query method responds the request addressed and returns the value of attribute. It takes one parameter, which specifies the name of the attribute, and returns the value of the attribute. If the value is not available, the internal Solution method is called and the domain agent tries to solve it in terms of its knowledge. If a solution is successfully achieved, the value will be returned, else it will request its relationships for the value of the attribute. If the value can be determined by the relationships, the value will be returned, else a value of -1.0 is finally returned. The algorithm of the Solution method is defined as follows:

```
i)      add the attribute to the **solve-pending-list**;
ii)     Search the **Kc** list, if **Kc.descrip.ad** equals the attribute, add to the K_list;
iii)    If  K_list is empty, go to ix;
```

iv) Set **Ck** to the head of **K_list**, create **dd_list**, and add all elements in **Ck.descrip,dd** to **dd_list**;

v) if dd_list is empty, go to step vii;

vi) set **dd** to the head of **dd_list**, if **dd** is in the **solve-pending-list**, go to step iii; else set **value** = **Query (dd)** ; if **value** = "-1.0", go to step iii; else set the **value** of **dd**, go to step v;

vii) evaluate Ck.descrip.formula, and set the value of the attribute, record the solution step, and return the value;

viii) Seach the **Rc** list, add all elements to the **Rc_list**;

ix) If Rc_List is empty, return "-1.0"

x) Set Rch to the head to Rc_list, if **value** = **Rch.Query ($name +"."+ attribute)**, if value = "-1.0", go to step ix; else and set the value of the attribute, record the solution step, and return the value.

xi) Stop.

- Decide: It is the interface to determine or set the value of a particular attribute.
- Explain: It explains how the value of an attribute is achieved. If the value is given, it returns GIVENContext, which is a particular tutorContext, if the value is not available, it returns NULL, else it returns a tutorContext explaining how the value is achieved. The structure of a tutorContext is discussed in detail in Chapter 6.
- RegistRel: It is the interface for registering an association with the instance of the class.

The Atom and Molecule classes inherit from the DomainAgent class. The Atom class possesses knowledge about the atoms, for example, the atomic weight of atomic elements. The Molecule class possesses knowledge about the molecules, for example, how to calculate molecular weight.

The Substance class models chemical substances. It inherits from DomainAgent, and has attributes like weight, volume, density, and the knowledge of the regularity among weight, volume and density. The Solid class inherits from Substance class. The Water class inherits from Compound class. It also has attributes like temperature, knowledge of its molecular formula, and the regularity between its density and temperature.

The Acid solution and Base solution class inherits from Electrolyte class. The Acid solution class defines attributes like molarity, number of H+ produced from each molecule of acid, W/V%, etc., it also possess the knowledge that regulates these attributes. The Base solution class defines attributes like molarity, number of OH- produced from each molecule of base, W/V%. It also possesses the knowledge that regulates these attributes. NaOH, KHP, FormicAcid and NicotinicAcid inherit from the Base or Acid class, respectively. They also have specific attributes and knowledge to themselves.

3.3.2 Relationships

There are many kinds of regularities in the General Chemistry domain. In our prototype system, we have defined four kinds of relationship classes, they are Association, Chem-Alias, Contain, and Neutralization. The Association class is an abstract based class for all relationship classes in the relationship class, and the Chem-Alias, Contain, and Neutralization models specific regularities.

Association

The Association class provides a template to model chemistry regularities. It is defined as follows:

```
        Association=( Association   (
                                         (string name)
                                         (string definition)
                                         (string extension)
                                    )
                                    (
                                    )
                                    (
                                         (DECIDE() string, string, void)
                                         (QUERY(), string, string)
                                         (EXPLAIN(), string, tutorContext)
                                         (AddObjects(), DomainAgent, void)

                                    )
                                    (
                                    )
                    )
```

The Association has three attributes: name, definition, and extension, which give general information about an association class. Four methods are also defined and explained as follows:

- Query: Query method responds the request addressed and returns the value of the object's attribute. It takes one parameter, which specifies the name of the Object's attribute, and returns the value. Since different kinds of regularities have different function, different association classes have to implement this method differently.
- Decide: It is the interface to determine or set the value of a particular attribute.

- Explain: It explains how the value of an object's attribute is achieved. If the value is not available, it returns NULL, else it returns a tutorContext explaining how the value is achieved. The structure of tutorContext is discussed in detail in Chapter 5.
- AddObjects: It is the interface for registering an object with the instance of an association class

Chem-Alias

Properties of chemical substances are either ***extensive*** properties or ***intensive*** properties. The value of an extensive property depends on *how much* matter is being considered. Mass, length and volume are all extensive properties. More matter means more mass, Another important point to note is that the value of extensive quantities can be added together. The value of an extensive property depends on the quantity of the matter. The measured value of an intensive property on the other hand does not depend on how much matter is being considered. For example, density and temperature are intensive properties. Suppose you have two equal quantity of water, each at exactly the same temperature. If you combine these two amounts the temperature of the combined water body would still remain the same, unlike mass or volume,.

Chem-alias refers to those instances that are initialized from the same object and have the same intensive property values. For example, in an acid titration experiment, you first standardize some sodium hydroxide solution, then you make several object aliases from this standardized sodium hydroxide to standardize other solutions. These instance aliases may be different in their weight, volume or other extensive properties, but they all have the same with the intensive properties, like molarity or density.

The Chem-alias class inherits from the Association class, and implements the Query method. When a request for the value of an object's attribute is received, it determines if the attribute is an extensive one, if not, it responds with "-1.0", else it initiates a request to other object on the specific attribute. If a value is finally achieved, it returns the value, else it returns "-1.0".

Contain

The Contain association class models the regularities that when several substances are physically put together in a container. It implements the following observations: 1) The volume of the container equals sum of the volume of the substances that fully fill the container; 2) the weight of the mixture is the total weight of all substances in the relationship.

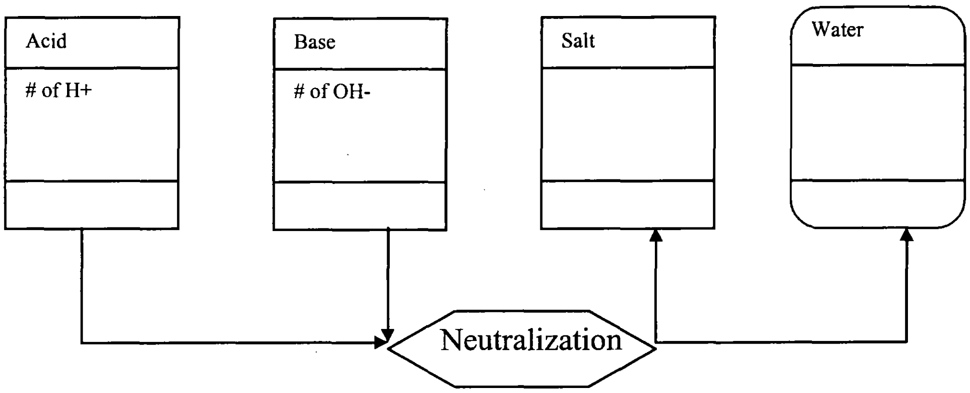


Figure 3.6 Neutralization Relationship Class

Neutralization

When an acid and a base are put together, neutralization happens and water (H_2O) and salt are produced. Figure 3.6 shows such chemical dynamic. The Neutralization class inherits from the Association class and implements in its Query method with the chemical regularity that when an acid neutralizes a base, the moles of H+ must be equal to the moles of OH-.

3.3.3 An Example Model for Acid Titration Laboratory

In acid titration laboratory, students learn how to standardize an acid or a base, and how to use a standardized solution to analyze other unknown base or acid solutions. A typical experimental procedure uses the following steps:

1. Standardize some NaOH solution using potassium hydrogen phthalate(KHP),
2. Then pipet out a unknown vinegar and titrate it with the standardized NaOH,
3. Finally weigh out a tablet containing a monoprotic acid,
4. Dissolve it and titrate it with NaOH of 1).

Students measure various data in this procedure, and are asked to analyze some of the properties of unknown solutions. For example, the following data are observed and unknown properties to be analyzed are:

Given:
The Mass of "KHP" weighed out (g) 0.71862
Volume of NaOH to neutralize "KHP" in 1) (ml) 12.4
Volume of Vinegar titrated (ml) 3.8
Volume of NaOH needed to neutralize Vinegar (ml) 16.0
Volume of NaOH needed to neutralize Tablet (ml) 8.6
MW of monoprotic acid in TABLET (g/mol) 145.399

Answer the following:
The mw of "KHP" is (g/mol)
The MOLES of "KHP" used is
The MOLARITY of the NaOH is (mol/Litre)
The MOLARITY of the VINEGAR is (mol/Litre)
The mg of ACID in the tablet is (mg)

During this typical procedure, objects involved are KHP, Vinegar, Tablet Solution, NaOH1, NaOH2 and NaOH3. KHP neutralizes NaOH1, NaOH2 neutralizes

Vinegar, NaOH3 neutralizes Tablet solution, and NaOH1, NaOH2 and NaOH3 are alias for the standardized NaOH solution. Figure 3.7 models the above typical procedure in acid titration laboratory.

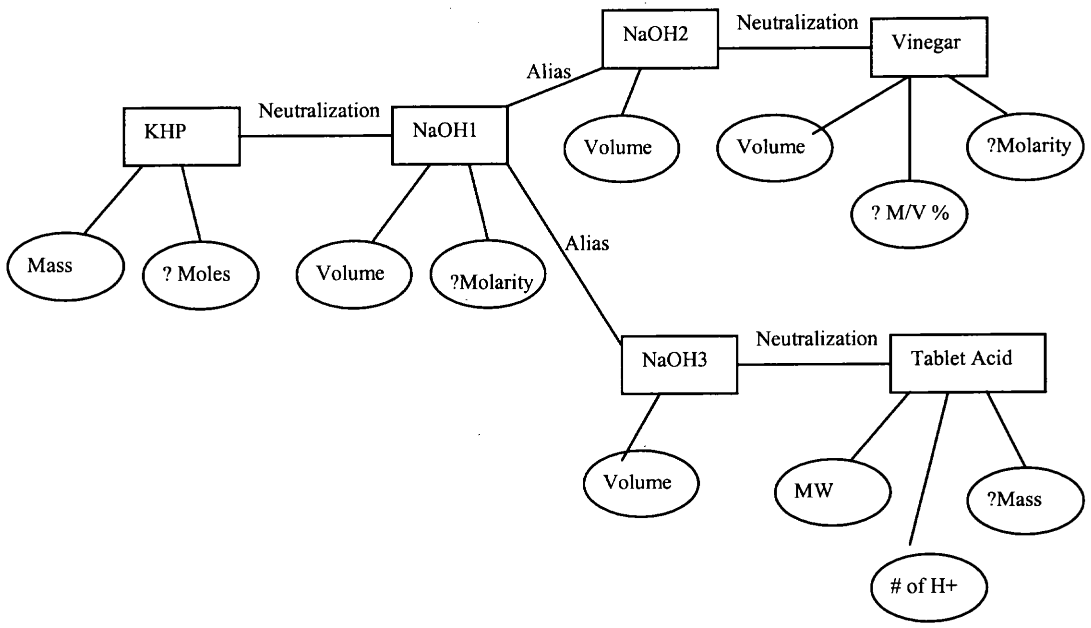


Figure 3.7 A Typical Object Model for Acid Titration Laboratory

3.4 The Knowledge Objects: A CORBA Implementation

3.4.1 The IDL Mapping of the Object Model

In DISTILS, all classes and relationship classes are implemented as CORBA-based knowledge objects. The interface of each knowledge object is defined using CORBA IDL.

The mapping from the object model to the IDL is quite straightforward. The attributes are mapped to IDL attributes, and the methods are mapped to IDL methods.

Since there is no explicit knowledge description in IDL, we embodied the knowledge into implementation of IDL methods. Figure 4.9 shows a partial IDL definition in DISTILS.

3.4.2 Cooperative Problem Solving of the Knowledge Objects

Each knowledge object is an encapsulation that is capable of deciding its own attributes, accepting and responding requests on its own attributes, and having social awareness to other knowledge objects. It is essentially an intelligent object.

Take NaOH as an example of a knowledge object. NaOH inherits from Base, which indirectly inherits from DomainAgent. NaOH therefore inherits all of its superclasses. NaOH owns its knowledge, both factual and procedural. Factual knowledge includes its molecular formula, its number of OH-, etc.. Procedural knowledge includes regularities like the dependency among molarity, moles and volume, the dependency among moles, weight and mass, etc.. NaOH could also know its relationships with other DISTILS knowledge objects.

The knowledge objects are capable of cooperating together to solve a particular problem. Take the example in section 3.3.3 as an example. 12 knowledge objects are initialized. KHP, NaOH1, NaOH2, NaOH3, Neu1, Neu2, Neu3, AliaChem1, Vinegar, Tablet, Atom, and Molecular are instance of KHP, NaOH1, Neutralization, AliaChem. Vinegar, Acid, Atom, Molecular class respectively (Figure 3.8).

In this example, the third question asks the Molarity of NaOH1. When a message (?molarity) is received by NaOH1, the NaOH1 object tries to solve it and searches for the value of its moles. However, the value of moles is not available and could not be determined by itself. NaOH1 turns to its relationship—neu1. The neu1 relationship tries to determine the value of NaOH1's moles. That in turn sends a message (?moles) to KHP object. KHP determines the value of its moles and returns it. The value of NaOH1's moles is then determined by neu1. Finally, NaOH1's molarity is determined.

3.4.3 Reusability of Knowledge Objects

The CORBA based encapsulation gives DISTILS knowledge objects the ability to be reused. In the prototype system of DISTILS, these objects are easily reused in different laboratories and are able to cooperatively solve various exercises. Figure 3.10 shows some exercises in DISTILS. Given that these objects have been developed under CORBA standard, they could also easily reused in other similar applications.

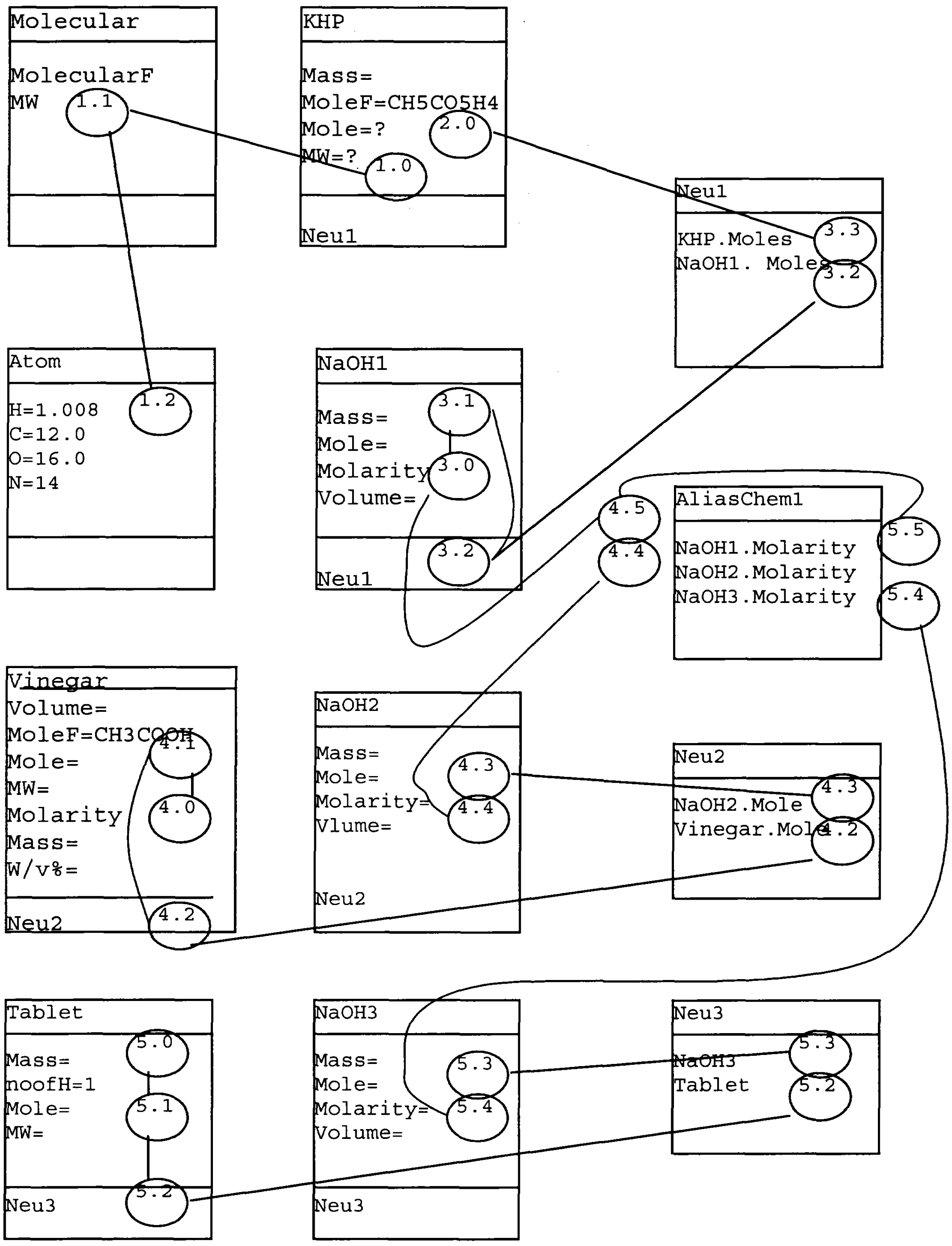


Figure 3.8 Cooperative Problem Solving of Knowledge Objects

```
interface tutorContext ;
interface Relationship ;
interface DomainAgent {
        readonly attribute string name;
        readonly attribute string definition;
        readonly attribute string extension;
        // primary methods of getting topic value
        string QUERY( in string topic ) ;
        void DECIDE( in string topic , in string value ) ; // setting topic value
        // explain the process of finding the answer
        tutorContext EXPAIN( in string topic ) ;
        //Register Relationships the agent invloved
        void RegisterRelationsips in Relationship ao ) ;
};
interface Substance : DomainObject {
        readonly attribute double density ;
        readonly attribute double volume;
        readonly attribute double weight ;
};
interface Particle : DomainObject {
};
interface Atom : Particle {
};
interface Molecular : Particle {
   readonly attribute string MoleFormular;
};
interface Base : Electrolyte {
   readonly attribute short noofOH ;
};
interface NaOH : Base {
};
interface Association {
        readonly attribute string name;
        readonly attribute string definition;
        readonly attribute string extension;
        string QUERY( in string topic ) ;
        void DECIDE( in string topic , in string value ) ; // setting topic value
        tutorContext EXPAIN( in string topic ) ;
         void addObjects( in DomainAgent obj );
};
interface Neutralization : Association {
};
interface AliasChem : Association {
};
```

Figure 3.9 A Partial IDL Definition in DISTILS

```
Exercise 1
(problem
(statement "Acid Titration Lab / A research chemist isolates a sample of nicotinic acid
(HC6H4NO2),  To determine it purity, /she  titrates some of the sample with
standardized NaOH solution./ The reaction is :
HC6H4NO2 + OH- ---> H2O + C6H4NO2 / / "
)
(objects
(Nicotinic Nicotinic )
(NaOH "NaOH" (Volume 43.6F "The volume of the standardized NaOH (ml) is " )
(Molarity 0.13F "The molarity of the standardized NaOH is (mol/Litre)" ) )
)
(associations
(Neutralization    (Neu1) (NaOH) (Nicotinic) )
)
(questions
(Nicotinic "Weight"  "The mass of Nicotinic acid in the sample (g) is")
)
)

Exercise 2
(problem
(statement "pH and Buffer / A weak monoprotic acid is weighted out, and diluted to
50.00ml. Exactly half of this is titrated with / NaOH until a pale pink phenolphthalein
end point is reached. This titrated solution is mixed with the unititrated / half of the
original solution and the pH of the mixture is measued. From the / following data
answer the questions./ /"
)
(objects
(Acid Acid (Weight 0.612F "Mass of the acid in grams" ) (noofH  1) (pH 2.87F "pH
of the original acid solution") (Volume 50.00F)  )
(NaOH "NaOH" )
)
(associations
(TitratedMixture ("half titrated mixture" (a 0.5F ) (b 1.0F) (pH 4.76F "pH of half-
titrated mixture") ) (Acid) (NaOH))
)
(questions
("Acid" pKa "What is the pKa for this acid")
("Acid" molarity "What is the molarity of the original solution?")
(Acid "MolecularWeight"  "What is the molecular weight of the acid"  )
)
)
```

Figure 3.10 Reuse in DISTILS exercises

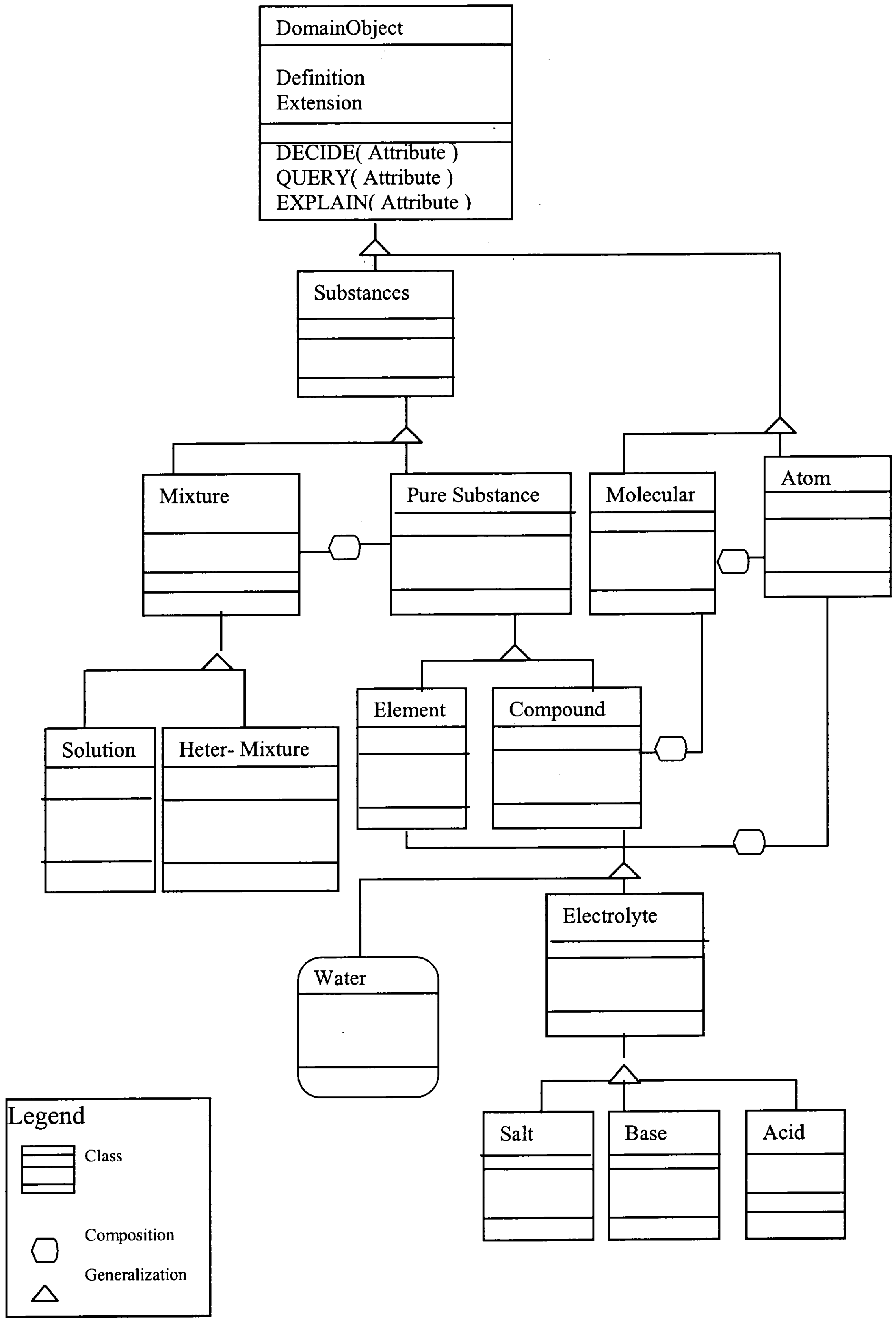


Figure 3.11 A Partial Diagram for DISTILS Object Model

CHAPTER 4

STUDENT MODELER

4.1 Student Modeling Techniques

Most existing ITSs developed a student model while trying to provide intelligent and effective tutorial activities, such as adaptive curriculum planning, behavior diagnosis, performance evaluation and tutoring motivation. Student model represents what the teacher understands about the student from their interactions. Student model includes two perspectives: personal attributes and knowledge model.

Personal attributes refer to the student's characteristics such as self-confidence, level of concentration, memory capability, and preferred presentation style, which are very important for effective tutoring. Milne (1996) presented an effort of development of the model of learner attributes and its use within adaptive tutoring system. In their work, multivariate statistical techniques were used to develop model of users' individual characteristics from empirical data. Matsubara (1996) proposed a human model which represents the student's internal psychological state and constructed a motivation system for the student in his/her learning process, to give the appropriate encouragement, praise or reproach messages.

Student knowledge model expresses what knowledge the student has mastered and what the student is still having difficulty with or has not uncovered. Overlay model was historically the oldest approach (Goldstein 1982). It represents the student's knowledge as the subset of an expert's knowledge. Its disadvantage is that it assumes that students' errors do not come from anomalies (bugs) in their knowledge, but only and always from incomplete knowledge. Such an assumption is unrealistic, as demonstrated by the inability of overlay models to explain students' behavior in a number of situations (Mitrovic 1996). Despite of its disadvantages, overlay model is still widely used in ITS applications because of its simplicity and ease of development.

Research has found that student modeling is so difficult that it is impractical to develop an accurate student model. It was also agreed that ITS could benefit substantially from even an inaccurate student model (Self 1994; Mitrovic 1996). DISTILS also benefits from a simple student model, which is described in following sections.

4.2 The Components for DISTILS Student Model

In DISTILS, To represent student knowledge status, an overlay model was developed. It includes two components: knowledge practice database and performance predictor.

4.2.1 Knowledge Practice Database

Each entry in knowledge practice database is a record of the student's experience of a knowledge unit and can be represented as the following tuple:

<KU, DL, TI, AR >

where:

KU denotes the name of a particular knowledge unit,

DL denotes the difficulty level of applying this knowledge in this practice, In DISTILS, the value of DL was currently set to 1.0 since only a limited number of questions were developed. However, DL is defined for future extensibility.

TI represents the time interval between previous practice and the current practice, and AR denotes the results of this knowledge practice, a value of 1 means a correct application while a value of -1 means a failed application.

ACID TITRATION LAB:

1) You standardize some NaOH solution using potassium hydrogen phthalate.
2) You then pipet out a vinegar unknown and titrate it with the standardized NaOH
3) Finally you weigh out a tablet containing a monoprotic acid.
4) You dissolve it and titrate it with NaOH of 1).

The Mass of "KHP" weighed out 0.7347
Volume of NaOH to neutralize "KHP" in A) 17.29
Volume of Vinegar titrated 5.00
Volume of NaOH needed to neutralize Vinegar 16.48
MW of monoprotic acid in TABLET 166.0
Volume of NaOH needed to neutralize Tablet 15.29

Calculate the Following

The mw of "KHP" is
The MOLES of "KHP" used is
The MOLARITY of the NaOH is
The MOLARITY of the VINEGAR is
The WEIGHT/VOLUME %VINEGAR is
The mg of ACID in the tablet is

Figure 4.1 A Sample Problem

Knowledge practice entries are created during the student's problem solving processes. For example, consider the problem in Figure 4.1. In this problem, the difficulty levels of these questions were assumed to be 1.0. The required knowledge for solving this problem includes neutralization, the atomic weight of K, H, C and O, KHP's molecular formula, vinegar's formula, the relation among molarity, moles and volume, and the relation among mole, molecular weight and mass. If a student solves these questions correctly, the student is assumed to have a successful practice with all the knowledge required, and the student knowledge practice database is shown in Table 4.1(a). Suppose three days later, the student fails in another question of difficulty level 1.0 in which knowledge Solution.Molarity-MW-Mass%, Solution.Molarity-Volume-Mole, and PureSubst-ance.Mole-MW-Mass are involved. The knowledge practice database will be updated as shown in Table 4.1(b). In Table 4.1(b), the time intervals for the records in Table 4.1(a) were updated to 72 hours since those records happened three days ago.

4.2.2 Performance predictor

One important aspect in student knowledge modeling is the performance evaluation. In most ITSs, one of three values: the student knows, the student does not know, and not sure if the student knows or not, is associated to a particular knowledge unit. Such a coarse-grained measure is limited, especially in a problem-solving environment, the tutor needs to evaluate among several knowledge units to pick up the most suitable knowledge unit to teach (Reye 1996). In DISTILS, a simple mechanism was developed to predict a student's performance on a knowledge unit based on his/her past performance.

Table 4.1 Sample of Knowledge Practice Database.

Knowledge	Difficult level	Time Interval (hrs.)	Application Result
Neutralization	1.0	0	1
K.AtomicWeight	1.0	0	1
H.AtomicWeight	1.0	0	1
C.AtomicWeight	1.0	0	1
O.AtomicWeight	1.0	0	1
KHP . MolecularFormular	1.0	0	1
Vinegar.MW	1.0	0	1
Solution.Molarity-MW-Mass%	1.0	0	1
Solution . Molarity-Volume-Mole	1.0	0	1
PureSubstance . Mole-MW-Mass	1.0	0	1

(a) Sample of knowledge practice database if a student solve the problem in Figure 4.1.

Knowledge object	Difficult level	Time Interval (hrs)	Application Result
Neutralization	1.0	72	1
K.AtomicWeight	1.0	72	1
H.AtomicWeight	1.0	72	1
C.AtomicWeight	1.0	72	1
O.AtomicWeight	1.0	72	1
KHP . MolecularFormular	1.0	72	1
Vinegar.MW	1.0	72	1
Solution.Molarity-MW-Mass%	1.0	72	1
Solution . Molarity-Volume-Mole	1.0	72	1
PureSubstance . Mole-MW-Mass	1.0	72	1
Solution.Molarity-MW-Mass%	1.0	0	-1
Solution . Molarity-Volume-Mole	1.0	0	-1
PureSubstance . Mole-MW-Mass	1.0	0	-1

(b) Sample of updated knowledge practice database

The student's past performance on a knowledge unit is first accumulated. It is the sum of the student's all past practice records in the knowledge practice database.

$$A = \sum_{i=1}^{N} AR_i \times DL_i \times e^{-\beta \times (TI_i - \frac{1}{TI_i})} \qquad (4.1)$$

In equation 4.1, AR_i is the result of the i^th past practice, DL_i is the difficult level, TI_i is the time interval from the practice to now, N is the number of records, and β is a tuned parameter. Equation 4.1 is simple but reasonable. The impact of a past practice record is determined by $AR_i \times DL_i$, the product of its difficulty level and application result. $e^{-\beta \times (TI_i - \frac{1}{TI_i})}$ is the time decay factor, which can be considered as the weight for each practice. The older a knowledge practice is, the less impact it has on the student's current performance, the smaller the $e^{-\beta \times (TI_i - \frac{1}{TI_i})}$ should be. Therefore, if the time interval, TI_i, of the i^th knowledge practice is very large, $e^{-\beta \times (TI_i - \frac{1}{TI_i})}$ tends to be zero, it means that the contribution of the i^th practice can be theoretically considered to be zero. However, if the time interval from a knowledge practice is very small, which means that the student practiced this knowledge a very short time ago, $e^{-\beta \times (TI_i - \frac{1}{TI_i})}$ is very large, and this practice dominates the prediction.

The prediction of a student's performance on a problem is based on the problem's difficulty level, DL_c, and his/her past performance. Based on ACT-R theory, the probability of a successful retrieval of knowledge chunks obeys a sigmoid function of the activation level (Anderson et al 1998). Equation 4.2 is used to predict a student's performance.

$$sp = \frac{1}{1 + DL_c \times e^{-A}} \qquad (4.2)$$

where: sp denotes the predicted student's performance, DL_c denotes the difficulty level of current practice, and A is the student's past performance calculated in equation 4.1

In DISTILS, β is determined in this way: given 3 positive practice in a day (24 hours), the past performance, A, should be above 2.4, which leads to a performance prediction above 0.9 according to equation (4.2); the influence of a practice a month ago (720 hours) should be consider as zero. In terms of these assumption, β is empirically determined to be 0.008. Table 4.1 shows the influence of selecting different β value. Table 4.1(a) shows that when β is set to 0.08, three positive practices in past 24 hour lead to a prediction value of 0.7651, while Table 4.1(b) shows that when β is set to 0.008, a prediction value of 0.9333 is achieved under the same situation. For more simulation results see Appendix C.

Table 4.2 The Influence of Different β Value

KU	DL	Ti(hrs)	Ar	Sp(β=0.08)
PS.Moles	1	24	1	
PS.Moles	1	24	1	
PS.Moles	1	2	1	
PS.Moles	1	0		0.7651

(a)

KU	DL	Ti(hrs)	Ar	Sp(β=0.008)
PS.Moles	1	24	1	
PS.Moles	1	24	1	
PS.Moles	1	2	1	
PS.Moles	1	0		0.9333

(b)

Table 4.2 shows some simulation results of the equation (4.2). Table 4.2 (a) and (b) shows that the older a practice is, the less influence it has on the current performance. The only difference in (a) and (b) is that Ar of the first practice of PS.Moles is different. However, since the first practice was done a long time ago, the performance difference predicted in (a) and (b) is very small. The only difference in (c) and (d) is that Ar of the last practice of PS.Moles is different. However, since the last practice was done a short time ago, the difference of performance prediction in (a) and (b) is significant.

Table 4.3 Some Simulation Results of the Equation (4.2)

KU	DL	Ti(hrs)	Ar	Sp(β=0.008)
PS.Moles	1	150	-1	
PS.Moles	1	43	-1	
PS.Moles	1	37	-1	
PS.Moles	1	12	1	
PS.Moles	1	5	1	
PS.Moles	1	0		0.58754

(a)

KU	DL	Ti(hrs)	Ar	Sp(β=0.008)
PS.Moles	1	150	1	
PS.Moles	1	43	-1	
PS.Moles	1	37	-1	
PS.Moles	1	12	1	
PS.Moles	1	5	1	
PS.Moles	1	0		0.6664

(b)

KU	DL	Ti(hrs)	Ar	Sp(β=0.008)
PS.Moles	1	150	-1	
PS.Moles	1	43	-1	
PS.Moles	1	37	-1	
PS.Moles	1	12	1	
PS.Moles	1	5	1	
PS.Moles	1	0		0.58754

(c)

KU	DL	Ti(hrs)	Ar	Sp(β=0.008)
PS.Moles	1	150	-1	
PS.Moles	1	43	-1	
PS.Moles	1	37	-1	
PS.Moles	1	12	1	
PS.Moles	1	5	-1	
PS.Moles	1	0		0.11385

(d)

Performance Prediction on Particular Knowledge Topic

As discussed chapter 3, a knowledge topic is defined as **T=(tid, TC, TR)**, where **tid** is the unique identifier of the topic, **TC** is a set of classes, and **TR** is a set of relationship classes. So the evaluation on a knowledge topic is comprehensively based the performance on **TC** and **TR**, and it is set to the average of the practice results.

Let

$\mathbf{tc}_i \in \mathbf{TC}$ (i = 1,.. m)

$\mathbf{tr}_j \in \mathbf{TR}$ (j = 1,.. n)

and $\mathbf{Ptc}_i$, $\mathbf{Ptr}_j$ represent the practice set of $\mathbf{tc}_i$, $\mathbf{tr}_j$ respectively.

$\mathbf{ptc}_{i,l} \in \mathbf{Ptc}_i$ (l = 1,.. p_i)

$\mathbf{ptr}_{j,k} \in \mathbf{Ptr}_j$ (k = 1,.. q_j)

The total number of practice on knowledge classes in this topic is:

$$A=\sum_{i=1}^{m} p_i$$

The accumulation of past performance on knowledge classes in this topic is:

$$PA=\sum_{i=1}^{m}\sum_{l=1}^{p_i} AR_{i,l}$$

The total number of practice on relationship classes in this topic is:

$$B=\sum_{j=1}^{n} q_j$$

The accumulation of past performance on relationship classes in this topic is:

$$\text{PB} = \sum_{j=1}^{n} \sum_{k=1}^{q_j} AR_{j,k}$$

then the evaluation of T can be calculated as follows.

$$et = \frac{PA + PB}{A + B} \qquad (4.3)$$

4.3 Student Modeler: CORBA-based Implementation

The student modeler is designed as a CORBA object. Figure 4.2 shows its interface definition.

```
interface studentModel {
   struct knowledgeTrace {
      string traceDate ;
      string knowledgeName ;
      string traceResult;
   };
   struct knowledgeStatus {
      string knowledgeName ;
      float reliability;
   };

   readonly attribute string studentName ;

   void traceUpdate( in knowledgeTrace t ) ;
   void addKnowledge( in string knowledge ) ;
   float predict( in string knowledge ) ;
   float predictTopic( in string topic ) ;
};
```

Figure 4.2 IDL Definition of Student Modeler

Each student modeler models a particular student's knowledge status. It is uniquely identified by studentName. Its functional operations include:

- traceUpdate: traceUpdate is called by the intelligent tutoring tool (Chapter 5) when it detects a change in student knowledge status. For example, when the student successfully applies the knowledge or makes a mistake.
- addKnowledge: this operation adds knowledge to current modeling space. It permits student modeler gradually to enlarge its modeling space, which improves the system performance. When initiated, a student modeler usually has empty modeling space. The tutoring components could use addKnowledge to request student modeler to pay attention to a particular knowledge, in this way, the modeling space gradually increased.
- predict: this operation should be called when performance evaluation on a particular knowledge unit is needed. It implements equation (4.2).
- predictTopic: this operation should be called when performance evaluation on a particular knowledge topic is needed. It implements equation (4.3).

CHAPTER 5

INTELLIGENT TUTORING TOOL

5.1 The Structure of ITT

Figure 5.1 shows the structure of ITT in DISTILS. It is composed of three modules: **Current Problem Space, Blackboard** and **Coach Delivery**. Current problem space stores a pool of knowledge objects, which are initialized in terms of problem statements. These knowledge objects are expected to solve the questions collaboratively and to generate and put solution plan and activities on **Blackboard. Coach Delivery** reads the solution plan and the activities from the **blackboard,** and the sequences coaching topics in term of student knowledge status. In the following sections, we discuss in detail how these components work and coordinate together to produce an intelligent tutoring behavior.

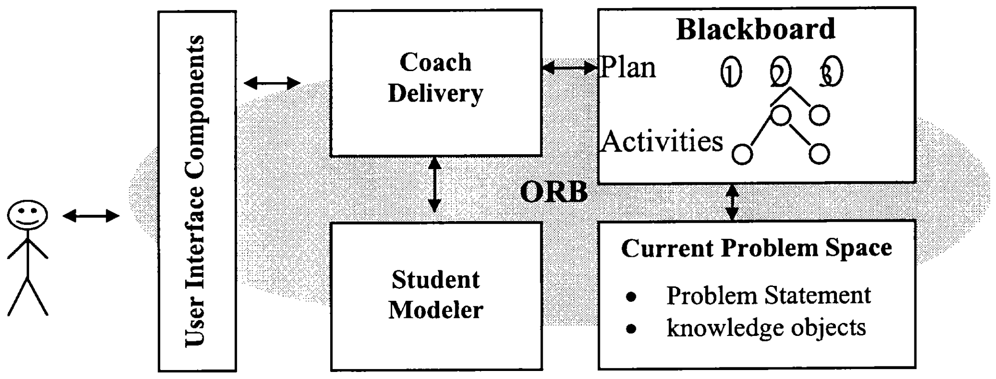


Figure. 5.1 Structure of ITT Module

5.1.1 Current Problem Space

Current problem space defines the current problem that a student is practicing. It consists of three components: problem statement, knowledge objects and their inter-relationships, and questions.

Problem statement is the description of the problem to be solved. For example, Figure 5.2 shows a sample problem statement for acid titration lab.

ACID TITRATION LAB:

1) You standardize some NaOH solution using potassium hydrogen phthalate.
2) You then pipet out a vinegar unknown and titrate it with the standardized NaOH
3) Finally you weigh out a tablet containing a monoprotic acid.
4) You dissolve it and titrate it with NaOH of 1).

Figure 5.2 Sample Problem Statement

As discussed in Chapter 3, an object model is applied to general chemistry domain. According to this model, we could formalize problems into knowledge objects, which interact. Each problem consists of several knowledge objects. Some attributes of these objects are initialized, and the objective is to find the value of other attributes of these objects. A Java-based symbolic expression mechanism (Detlefs 1999) is used to describe knowledge objects in chemical problems.

```
(KnowledgeClass     InstanceID
        (AttributeName1 AttributeValue Description)
        (AttributeName2 AttributeValue Description )
                              :
        (AttributeName-n AttributeValue Description )
)
```

KnowledgeClass identifies the type of this knowledge object and InstanceID uniquely denotes a knowledge object in this problem. The remaining lists are attribute lists that initialize the attributes for the knowledge object. The first element of each attribute list is the name of the attribute, the second one is the value for this attribute, and the third one is a description of the attribute. Figure 5.3 shows an example of description of knowledge object instances in a problem of Acid Titration Lab. In this example, three knowledge objects are to be initialized: NaOH, KHP, and Vinegar. The value of NaOH's Volume is set to 13.50, the value of KHP's weight is set to 0.6951, and the value of Vinegar's volume is set to 9.00, respectively.

```
(
        (NaOH  NaOH
                (Volume 13.50F "Volume of NaOH to neutralize 'KHP' in A)(ml)" )
        )
        (KHP   KHP
                (Weight 0.6951F "The Mass of 'KHP' weighed out(g)"
        )
        (Vinegar   Vinegar
                (Volume 9.00F "Volume of Vinegar titrated(ml)"
        )
)
```

Figure 5.3 Sample definition of knowledge objects

The interaction relationship among these instances can be described as follows:

```
(RelationName
        (InstanceName
                (Attribute-1 Value-1 Description-1)
                        :
                (Attribute-n Value-n Description-n)
        )
        (Role-1 )
        (Role-2 )
          :
        (Role-n)
)
```

RelationName identifies the type of the relationship. InstanceName uniquely denotes a the relationship. Attribute-1, Attribute-2, .., and Attribute-n define the attributes for this relationship, with Value-1, Value-2, .., Value-3 and Description-1, Description-2, .., and Description-n are their value and description. Role-1, Role-2, .., and Role-n are knowledge objects that involve in this relationship.

Questions are also formalized as follows:

```
(Questions
        (InstanceName-1     AttributeName-1)
        (InstanceName-2     AttributeName-2)
                     :
        (InstanceName-n     AttributeName-n)
)
```

The AttributeName-1 of knowledge object InstanceName-1, the AttributeName-2 of knowledge object InstanceName-2, .. , and the AttributeName-n of knowledge object InstanceName-n are the questions. Figure 5.4 gives an example problem space.

```
(problem
(statement "Acid Titration Lab / 1) You standardize some NaOH solution using
KHP./ 2) You then pipet out a vinegar unknown and titrate it with the standardized
NaOH /3) Finally you weigh out a tablet containing a monoprotic acid ./ 4) You
dissolve it and titrate it with NaOH of 1)./ The molecular formula of KHP is
KC8H5O4/ "
)
(objects
(NaOH  NaOH (Volume 13.50F "Volume of NaOH to neutralize 'KHP' in A)(ml)" )
)
(KHP  KHP (Weight 0.6951F "The Mass of 'KHP' weighed out(g)" ) )
(Vinegar  Vinegar (Volume 9.00F "Volume of Vinegar titrated(ml)" ) )
(NaOH  NaOH--2 (Volume 19.92F "Volume of NaOH needed to neutralize
Vinegar(ml)" ) )
(NaOH  NaOH--3 (Volume 20.08F "Volume of NaOH needed to neutralize
Tablet(ml)" ) )
(Acid  "Tablet Acid" (noofH 1) (MolecularWeight  145.8F "Molecular Weight of
monoprotic acid in TABLET(g|mole)") )
)
(associations
(Neutralization  (Neu1) (NaOH) (KHP) )
(Neutralization  (Neu2) (NaOH-2) (Vinegar) )
(Neutralization  (Neu3) (NaOH-3) ("Tablet Acid") )
(AliasChem  (AliasChem1) (NaOH) (NaOH-2) (NaOH-3) )
)
(questions
(KHP MolecularWeight "The molecular weight of 'KHP' is (g/mol)")
(KHP Moles "The Moles of 'KHP' is" )
(NaOH Molarity "The Molarity of NaOH is (mol/Litre)")
(Vinegar Molarity "The Molarity of Vinegar is (mol/Litre)")
 ("Tablet Acid" Weight "The Mass of Acid in the Tablet (g)" )
)
)
```

Figure 5.4 Sample Problem Space

5.1.2 Blackboard

Blackboard is a mechanism to share information among multiple knowledge sources (Jagannathan V., Dodhiawala R., Baum L.S., 1989). In DISTILS, each knowledge object represents a knowledge source. A blackboard mechanism is adopted to record the cooperative problem solving processes by knowledge objects. It consists of two components: plan and activities. Its structure is defined as follows:

BB = (BBid, Plan, Activity)

- BBid is the unique identifier of the blackboard. Usually each ITT has its own blackboard, so usually BBid is set to the same identifier as the ITT.
- Plan is a set of bipartite, (**oid, attrname**). **Oid** is the object identifier of a knowledge objects in current problem space, **attrname** is the name of the attribute of the knowledge object to be solved.
- Activity is a set of tuples, (**givens, question, explain, kid**). The **givens** represents the known fact involved in this activity, the **question** sets the goal to solve, the **explain** provides explanation how the solution is achieved, and the **kid** refers the knowledge embodied in this activity. The **givens** is a set of items, and the **question** is represented as one item. An item is a tuple, (**objId, topic, strValue, dValue, valueType, itemType, kid**). **objID** refers the object this item belongs to, topic is the attribute name. **strValue** and **dValue** are the value of the item in string and double type, respectively. **valueType** switches the value type, with 0 means double type and 1 means string type. **itemType** denotes if this item is given data, or factual knowledge or a question, with value of 0,1,2 refers given data, knowledge or question, respectively. **kid** refers the knowledge if itemType equals to 1.

Instructional plan is a sequence of tasks toward solving the problem. When human teachers try to solve a problem, they always set up a solution plan with correct precedence relations for step by step process, which is very effective in both problem-solving and tutoring.

The plan generation process is difficult and time-consuming. It is not the research focus of DISTILS. Instead, in DISTILS, we are using solution plan, which is provided by human teacher or problem provider, to guide the cooperative problem solving process by knowledge object instances. Each question becomes a phase in the solution plan, for example, questions in Figure 5.4 can be represented as the following solution plan (Figure 5.5).

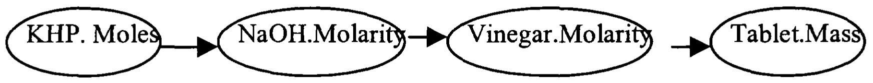


Figure 5.5 A Solution Plan

Each phase in the solution plan is decomposed into a collection of related activities. Each activity may be decomposed into a collection of related sub - activities. Therefore, for each phase in the solution plan, an activity tree is derived. Figure 6.6 shows the activities involved in the solution to phase 1 and 2 showed in Figure 6.5. These activities in solution phase 1 can be formatted as follows:

```
(

        (
                (//givens
                        (KHP Mass 0.6951 0.6951 0  0 null)
                        (KHP MolecularWeight 204.23 204.23 0  0 null)
                )
                (KHP Moles 0.0034 0.0034 0  0 null)
                "Moles = Mass / Molecular Weight"
                PureSubstance.Moles
        )
        (
                (//givens
                    (KHP MolecularFormula  KC8H5O4 1  0 null)
                    (K AtomicWeight 39.0 39.0 0  1 K.AtomicWeight)
                    (C AtomicWeight 12.0 12.0 0  1 C.AtomicWeight)
                    (O AtomicWeight 16.0 16.0 0  1 O.AtomicWeight)
                    (H AtomicWeight 1.003. 1.003 0  1 H.AtomicWeight)

                )
                (KHP MolecularWeight 204.23 204.23 0  0 null)
                "Molecular Weight of KHP is calculated as 39+8*12+16*4+1.003*5"
                Molecular.MWCalculation
        )

)
```

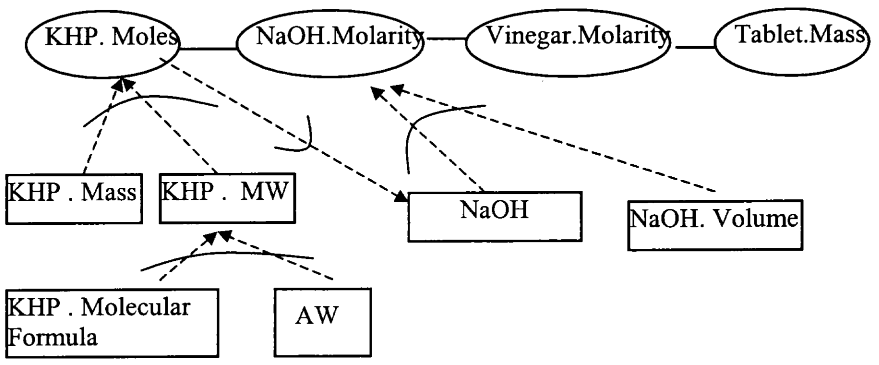


Figure 5.6 Activities in A Solution

5.1.3 Coach Delivery

Coach delivery is the kernel of ITT. It comprehensively collects information from student modeler and blackboard to produce intelligent tutoring behavior. During this process, the activities that are related to current solution phase are first collected, then a mechanism to describe the knowledge context in current solution plan is established, finally continuous tutoring interactions are delivered according to a particular tutoring situation. The algorithm for collecting activities, establishing knowledge context, and tutoring delivery is described in the following sections.

- activityCollection Algorithm

Input: cps: current solution phase
Output: tc: collection of activities related to cps

```
1. set cp to current solution phase, tc=null
2. set vector=null, add cp to vector;
3. if vector == null, go to step 11
4. set head = vector . elementAt(0) ;
5. search the blackboard for the activity, act, that satisfies act. objId=head.oid and
   act. topic =head. attrname. If act==null, go to step 11
6. add act to the head of tc
7. set nItem = act. givens.length() ; I = 0;
8. if( I >= nItem ) go to step 10
9. if act.givens[I].itemType ==0 and act.givensI] is not a question or given data in
   current problem space, set cp.oid = act.given[I].objID and attrname=
   ct.givens[I].topic, add cp to vector, I=I+1, go to step 8
10. remove the head of vector, go to step 3
11. stop
```

- knowledgeContextTree Algorithm

Input: tc
Output: kct: knowledge context tree

The output is a knowledge context tree, in which an array of a data structure, kldgContext, is defined to describe the knowledge context in a solution phase as follows.

```
class kldgContext {
  String KID;        // the knowledge name applied
  int nStep ;        // The activity this knowledge applied
  int nItem ;        // The item this knowledge applied in the activity
  float reliability ; // the predicted student knowledge performance
  int[] child ;      // The activities prerequisite to this knowledge applied
  int    parent ;    // The activity to which this knowledge applied is prerequisite
};
```

knowledge context tree describes knowledge application situation and their sequencing in the solution phase. For example, let us consider the solution phase 1 in Figure 6.6. In this example, we have two steps. The first step is the calculation of KHP's molecular weight, the second step is the calculation of KHP's moles. The content of this knowledge context tree is described in Table 5.1, which is graphically shown in Figure 5.7.

Table 5.1 A Sample Knowledge Context Tree

	KID	nStep	nItem	Reliability	nChild	nParent
0	Molecular.MolecularWeight	1	0	0.50	1,2,3,4	5
1	K.AW	1	1	0.53	Null	0
2	O.AW	1	2	0.63	Null	0
3	C.AW	1	3	0.67	Null	0
4	H.AW	1	4	0.77	Null	0
5	PureSubsatnce.Moles	2	0	0.79	0	null

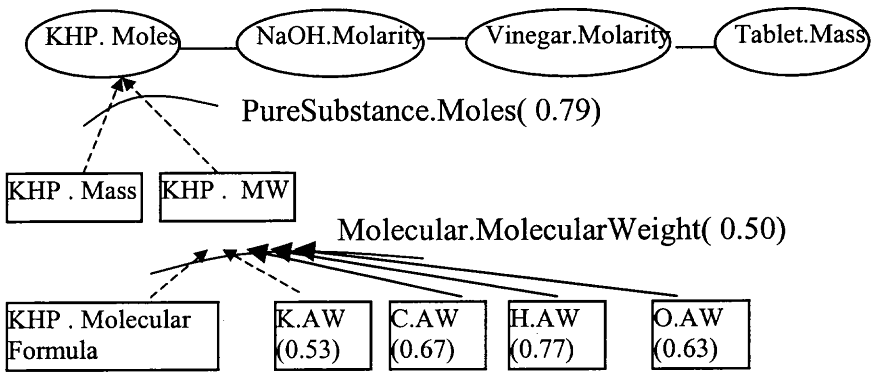


Figure 5.7 Activities with Knowledge Context

The algorithm for knowledge context tree is as follows:

1. set kct= null;
2. search all KIDs in the activities in tc, get performance evaluation from student modeler and sort them according to knowledge status under ascending order and add them to kct .
3. set nActivity = the no of activities in tc, j = nActivity-1;
4. if j < 0 go to
5. set act = tc.stepAt(j) ;
6. set nItem = act. givens.length() ; I = 0;
7. if(I >= nItem) go to

8. if **act.givens[I].itemType ==0** and **act.givens[I]** is not a question or given data in current problem space, find **kth** activity, act1, where **act1.question.objId= act.givens[I].objId and act1.question.topic= act.givens[I].topic.** link knowledge context j,0,k,0.
9. if **act.givens[I].itemType ==1**, link knowledge context (j,0, j, I+1)
10. I = I + 1, go to step 7
11. j=j-1, go to step 4.
12. stop.

- TutorNext Algorithm

TutorNext algorithm takes one input parameter, bPrev. If the student performs correct in the previous interaction, bPrev=true, else bPrev=false. When tutorNext is called for the first time, bPrev=false. The algorithm is as follows:

1. set **cur=** the knowledge that applied in previous tutoring interaction. **Cur** is set to null before the first time tutorNext is called.
2. if **bPrev= false,** if **cur != null,** update student model. Set **p = findSmallest(cur)** . if **p!=null, cur=p, return tutor(p);**
3. else if **cur != null,** update student model. **RemoveSubtree(cur). Set Cur=null,** return tutorNext(false);
4. stop

In the above algorithm, findSmallest(), removeSubtree(), and tutor() are auxiliary functions. Function findSmallest finds the knowledge context whose reliability is the smallest in knowledge context tree; Function removeSubtree(p) remove all the sub knowledge context whose parent directly or indirectly link to knowledge context p. Function tutor returns a tutoring structure related to the knowledge context.

5.2 ITT: A CORBA-based Implementation

The ITT in DISTILS is designed as a CORBA object, together with another CORBA object, tutorContext. Figure 5.8 shows important parts of their IDL definition.

5.2.1 The Interface tutorContext

The interface tutorContext provides a structure for describing the activities in the blackboard. It defines two structures: tutorItem and tutorScenario. The tutorItem structure is used to describe a given condition or question in an activity and the tutorScenario describes an activity.

In tutorItem, ObjID denotes which object it belongs and topic refers to the name of the attribute involved. valueType is the switch for the value of the attribute, with strValue and dValue store the double value or string value, respectively. itemType tells whether this item is a given condition or a knowledge or a question, and knowledgeID stores the knowledge when applicable.

The tutorScenario structure describes components involved in an activity. It consists of given conditions, the question, the explanation and the knowledge applied in this activity. The **givenOrKldg** is the sequence of tutorItems describing the pre-conditions. The **question** is the item to be answered, the **knowledgeID** denotes the knowledge applied and the **explain** stores the explanation that how the solution is achieved.The tutorContext basically is a list of steps toward a solution phase in the plan. Each step is represented as a tutorScenario. The interface provides a set of methods for managing the steps, such as adding, inserting, removing, and accessing a step activity.

```
interface tutorContext {
  struct tutorItem {
      string objID ;
      string topic ;
      string strValue ;
      double dValue ;
      short valueType ;   //0--double , 1 -str
      short itemType ;    // G/K/Q
      string knowledgeID ;
  };
  struct tutorScenario{
      sequence<tutorItem > givenOrKldg ;
      tutorItem question ;
      string topic ;
      string explain;
      string knowledgeID ;
  };
  attribute string topicName;

  void addStep( in tutorScenario s );
  void insertStepAt(in tutorScenario s ,in long index);
  void removeStep( in string topic ) ;
  tutorScenario nextStep() ;
  tutorScenario stepAt(in short i) ;
};

interface ITT {
  void setProblem( in string ques ) ;
  void setStudent( in string Student ) ;
  string problemStatement() ;
  string problemDescription() ;
  string givensAt(in long i ) ;
  string quesAt( in long i ) ;
  oneway void solve() ;
  boolean checkAns(in long index, in double myans) ;

  // get tutorScenario
  void setCurrentTutorTopic( in string topic ) ;
  tutorContext currentTutorContext();
  tutorContext::tutorScenario tutorNext(in boolean bPrev) ;
};
```

Figure 5.8 The IDL Definition of ITT

5.2.2 The Interface ITT

The interface ITT describes operations provided by a tutor. The operations are grouped in to 4 groups: initialization, access, solution, tutoring. The implementation of the four group operations are explained as follows:

I. Initialization

- setProblem: It reads the problem statement from the database through repository component. The problem is parsed and the problem space is initialized.
- setStudent: It locates the student modeler of the student. If the student modeler is not launched, a new one is created and launched.

II. Access

The access group operations is provided for the interface component to get necessary information to display a given problem.

- problemStatement: it returns the problem statement.
- problemDescription: it returns the problem description.
- givensAt: it returns the description of a given data in the problem.
- quesAt: it returns the description of a question in the problem.

III. Solution

- solve: it launches the cooperative problem solving processes of knowledge objects in the problem space. Each question in the problem is used as a solution plan phase and its solution activities are represented in the associated tutorContext.

IV. Tutoring

- setCurrentTutorTopic: it sets the current plan phase and is used to denote which plan phase needs to be taught.

- currentTutorContext: it returns solution activities associated with the current plan phase. It also communicates with the student modeler to get the student knowledge status on the knowledge involved each activity and produce a data structure for sorting these activities. It implements activityCollection and knowledgeContextTree algorithms.
- tutorNext: It returns the next activity to be discussed by implementing tutorNext algorithm. It takes one parameter, which tells the student's feedback in a previous activity. If the value of the parameter is true, it means that the student did the unit correctly, else, it means the student failed.

CHAPTER 6

ADAPTIVE LECTURE GUIDANCE

6.1 The Structure of Adaptive Lecture Guidance (ALG)

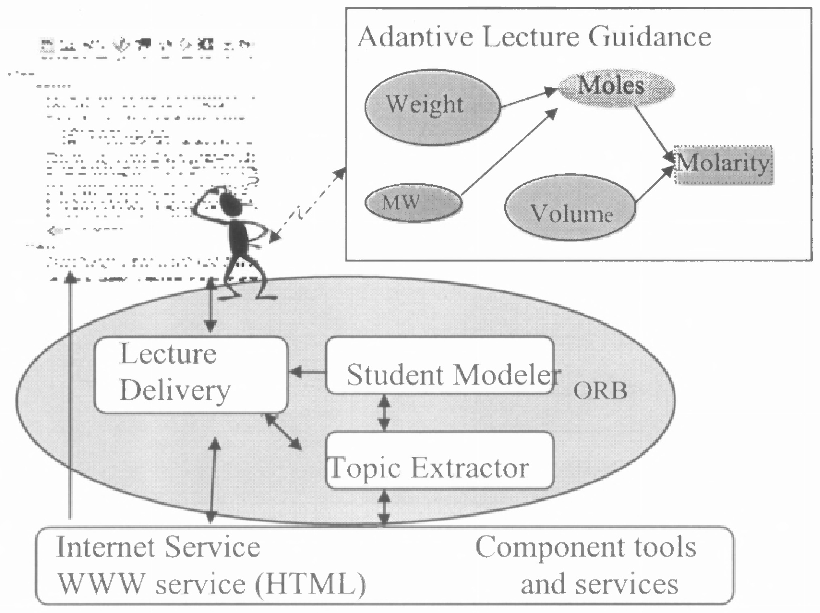


Figure 6.1 Structure of Adaptive Lecture Guidance

The structure of adaptive lecture guidance module is illustrated in Figure 6.1. The student modeler is created at the server side that evaluates student knowledge status independently. When the student gets confused or lost during navigation, the ALG can be helpful. The ALG consists of two components: topic extractor and lecture delivery. The topic extractor analyzes knowledge topics in the current page. The lecture delivery first contacts the topic extractor to get the HTML Learning Model (HLM), then it communicates with student modeler to get the student's knowledge status. Finally, it produces adaptive interface components illustrating where the student is and suggesting the next step.

6.2 Topic Extractor

To give adaptive guidance when students get lost, it is important to let students, first, to know where they are at first. So it is necessary to find a mechanism for describing the content of hypermedia lecture pages. In this section, we will discuss how knowledge elements in a Web page are modeled.

A HLM is associated with each HTML page for describing its educational contents. A generic HLM is defined as follows:

HLM = **(HLMid, KE, L)**

- **HLMid** is the unique identifier of the HLM. In DISTILS, we create a new URL by adding suffix '.xml' to the URL for original HTML page, and use the new URL as HLMid. For example, if the URL for original HTML page is http://bengu-pc2.njit.edu/distil/density/density.html, then the HLMid for the HML of this page is http://bengu-pc2.njit.edu/distil/density/density.html.xml.
- **KE** is a set of knowledge elements in the HLM. Each element of **KE** is tuple, **(keid, topic, class, element, url, cord). Keid** is the unique identifier in the HLM, **topic, class** and **element** specifies the topic, knowledge class and element of this knowledge. **cord** specifies the coordinates where the element display. **url** refers to the Web page that discuss this knowledge, a value of null means this knowledge is discussed in current page.

- **L** is a set of prerequisite relationships among elements in **KE**. Each element of **L** is a bipartite, **(keidpre, keidcon)**. Both **keidpre and keidcon** are among **KE**, and **keidpre** is prerequisite to **keidcon**.

The following is an example of HLM.

```
("http://bengu-pc2.njit.edu/distils/acidtitration/index.html.xml"
        (
                (Molarity, solution, solution, molarity, http://bengu-pc2.njit.edu/trp-
                chem/chemistry/solutions/Sol.html#4.13, (150,99))
                (Acid Titration, titration, null ,null,null, (184,25))
                (Neutralization, titration, neutralization, null, http://bengu-pc2.njit.edu/trp-
                chem/aspirins/nap5.html, (241,99) )
                (Acid, titration, acid, null, http://bengu-pc2.njit.edu/trp-chem/aspirins/nap5.html
                (300,174))
                (Base, titration, base, null, http://bengu-pc2.njit.edu/trp-chem/aspirins/nap5.html,
                (241,174))
                (Moles, MatterStructure, PureSubstance, Moles, http://bengu-
                pc2.njit.edu/distils/Acidtitration/awandmw.html, (189,174))
                (Substance, Matter, Substance,Measurement, http://bengu-pc2.njit.edu/trp-
                chem/chemistry/introdu/chemie1sm2.html,(116,258))
                (Molecular, Atomic Thoery, Molecular, MWCalculation, http://bengu-
                pc2.njit.edu/distils/Acidtitration/awandmw.html, ( 160,339))
                (Atom, Atomic Theory, Atom, AtomicWeight, http://bengu-
                pc2.njit.edu/distils/Acidtitration/awandmw.html, (271,339))
        )
        (
                (Molarity, Acid Titration) (Neutralization, Acid Titration) (Acid, Neutralization) (Base,
                Neutralization) (Moles, Molarity) (Substance-Molarity) (Molecular-Moles) (Substance-
                Moles) (Atom-Molecular)
        )
```

Figure 6.2 An Example of HLM

The main function of topic extractor is to analyze the HLM associated with each HTML page and format it into a well-defined data structure. It is defined as follows using CORBA IDL.

```
Interface topicExtractor {
enum color {unmastered, unknown, practiced, mastered, notprepared, encouraged };
enum shape {rectangle, roundRectagle};
struct KnowledgeNode {
   double x;  // the x coordinate to display this node
   double y; // the x coordinate to display this node

   color nodecolor; // the color for this node
   int noofFrom;     // the number of prerequisite knowledge of this knowledge
   int noofUnmasteredFrom ; // the number of unmastered prerequisite knowledge
   int shape ; //

   String lbl;  // the display label for this knowledge
   String url;  //the URL discussing this knowledge
   String knowledge; // the knowledge name in Topic.Class.Element format
};

struct Edge {
   int from;
   int to;

};
struct ALGGraph{
sequence<knowledgeNode> nodes ;
sequence<Edge> links ;
string url;

};

         ALGGraph extract(in string url);
};
```

The algorithm for extract method is as follows:

1. Open the HLM file named by parameter-url,
2. Read the HLMid, check if HLMid equals to the combination of url and ".xml", if not, return null,

3. Get the **KE** list, set **KEElem=** first element of **KE, KE** = remaining of **KE,**
4. If **KEElem==null**, goto step 6,
5. Parse **KEElem**, add new node to ALGGraph, set **KEElem=** first element of **KE, KE** = remaining of **KE,** go to step 4;
6. Get the **L** list, set **LElem=** first element of **L, L** = remaining of **L,**
7. If **LElem==null**, goto step 9,
8. Parse **LElem**, find the index of **keidpre** and **keidcon** in nodes, add new link to ALGGraph, set **LElem=** first element of **L, L** = remaining of **L,** go to step 7
9. Stop, return ALGGraph.

6.3 Lecture Delivery

Lecture delivery communicates Student Modeler and topic extractor for student knowledge status and educational contents of current HTML page, respectively, and presents adaptive guidance based on these two kinds of information. Lecture delivery uses different colors and shapes to illustrate the particular situation of the student. A rounded rectangle is used to show where the student is. A white box shows that the student demonstrated sufficient proficiency; a gray box means that the ALG knows nothing about the student on the topic; a yellow box means the student performance was satisfactory but more practice is needed; a green box encourages the student to practice; a red box suggests that the student is not ready for this topic.

After getting the ALG graph from topic extractor, lecture delivery verifies the student knowledge status with student modeler and applies the different colors to knowledge nodes in ALG graph. The algorithm for coloring ALG graph is as follows:

1. While there is a knowledge node, **node-1**, in ALG graph is not colored, set **n =node-1**, if all knowledge nodes are colored, go to step 3;
2. Communicates with student modeler and gets the knowledge status, **ks**, for **n.** If **ks** < 0.45, set n.nodecolor = unmastered, else if **ks** < 0.55, set n.nodecolor = unknown, else if **ks** < 0.7, set n.nodecolor = practiced, else n.nodecolor = mastered. Go to step 1;
3. For each link, **ln**, in ALGGraph, if ALGGraph.**nodes[ln.from].color** == mastered, ALGGraph.**nodes[ln.to].noofUnmasteredFrom --** ;
4. For each node, **n**, in ALGGraph.

```
if (n . color == mastered ) ;
else if( n.noofFrom != 0 ) {
      if( n.noofUnmasteredFrom == 0)
            n.color = encouraged ;
      else if(n.color == unmastered) //practiced, unknown color remain unchanged
         n.color = unexpected;
}
```

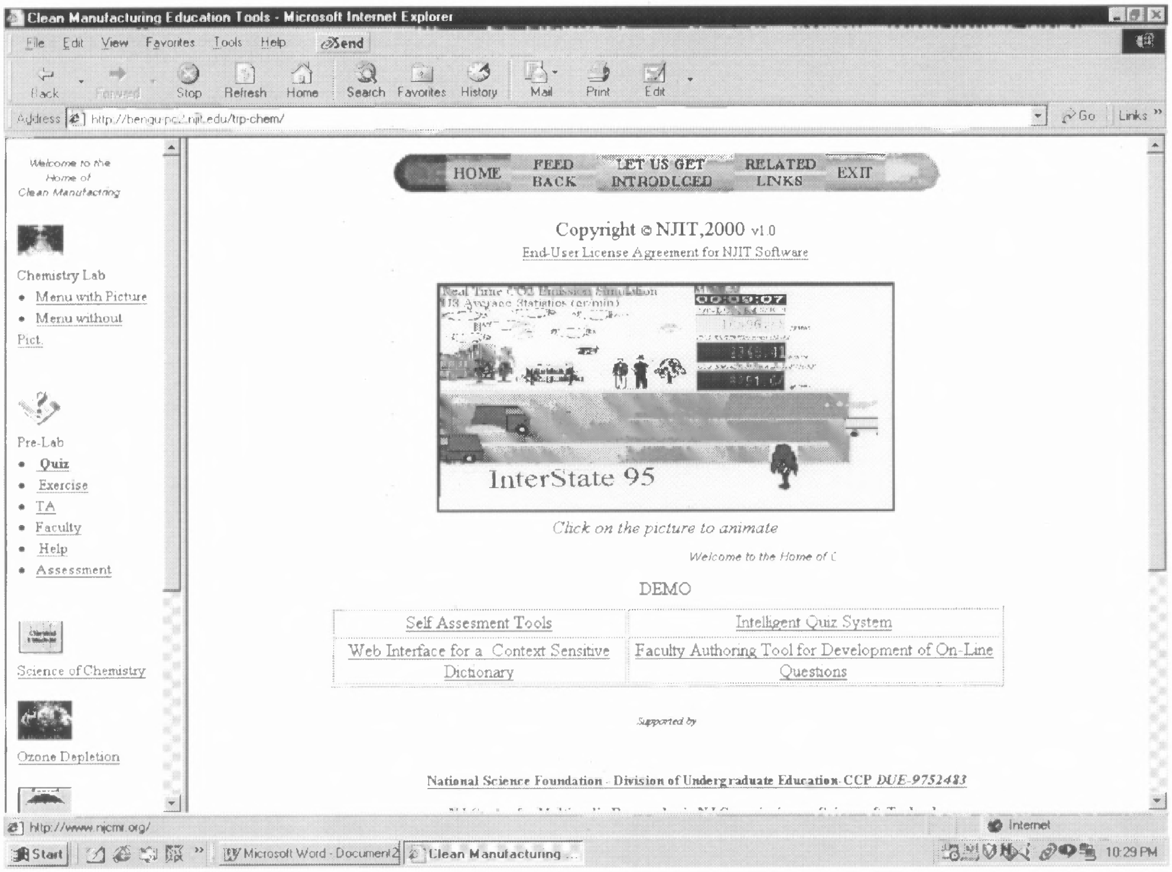
CHAPTER 7

IMPLEMENTATION AND EVALUATION

7.1 Implementation

7.1.1 The General Chemistry Laboratory Courseware

This prototype system has been implemented in the environment of general chemistry laboratory courseware developed at NJIT. Figure 7.1 shows the homepage of this entire courseware. The Chemistry Lab part is a Web-based multimedia laboratory manual. It discusses the safety issue, the background and other related issues for each laboratory experiment. Videos and animations were also provided to illustrate the procedure for each experiment. The general chemistry textbook material and related case studies are enhanced by student self-assessment tools, which are a series of pop-up questions for testing student's understanding of the materials. The Pre-lab part includes a quiz tool, a TA administration tool, a faculty tool and the DISTILS exercises. The quiz tool provides students with a pre-lab test that they must pass before going to the actual laboratory. The faculty authoring tool helps faculty members to prepare the pre-lab quiz. The DISTILS prototype system is developed to provide adaptive tutoring in this courseware.


Clean Manufacturing Education Tools - Microsoft Internet Explorer
File Edit View Favorites Tools Help Send
Back Forward Stop Refresh Home Search Favorites History Mail Print Edit
Address http://benqu-pc2.njit.edu/trp-chem/
Go Links
HOME
FEED BACK
LET US GET INTRODUCED
RELATED LINKS
EXIT
InterState 95
http://www.njcmr.org/
Internet
Start
Microsoft Word - Document2
Clean Manufacturing ...
10:29 PM

Welcome to the Home of Clean Manufacturing

Chemistry Lab
- Menu with Picture
- Menu without Pict.

Pre-Lab
- Quiz
- Exercise
- TA
- Faculty
- Help
- Assessment

Science of Chemistry

Ozone Depletion

Copyright © NJIT,2000 v1.0
End-User License Agreement for NJIT Software

Real Time CO2 Emission Simulation
US Average Statistics (gr/min)
00:09:07

Click on the picture to animate

Welcome to the Home of C

DEMO

Self Assesment Tools	Intelligent Quiz System
Web Interface for a Context Sensitive Dictionary	Faculty Authoring Tool for Development of On-Line Questions

Supported by

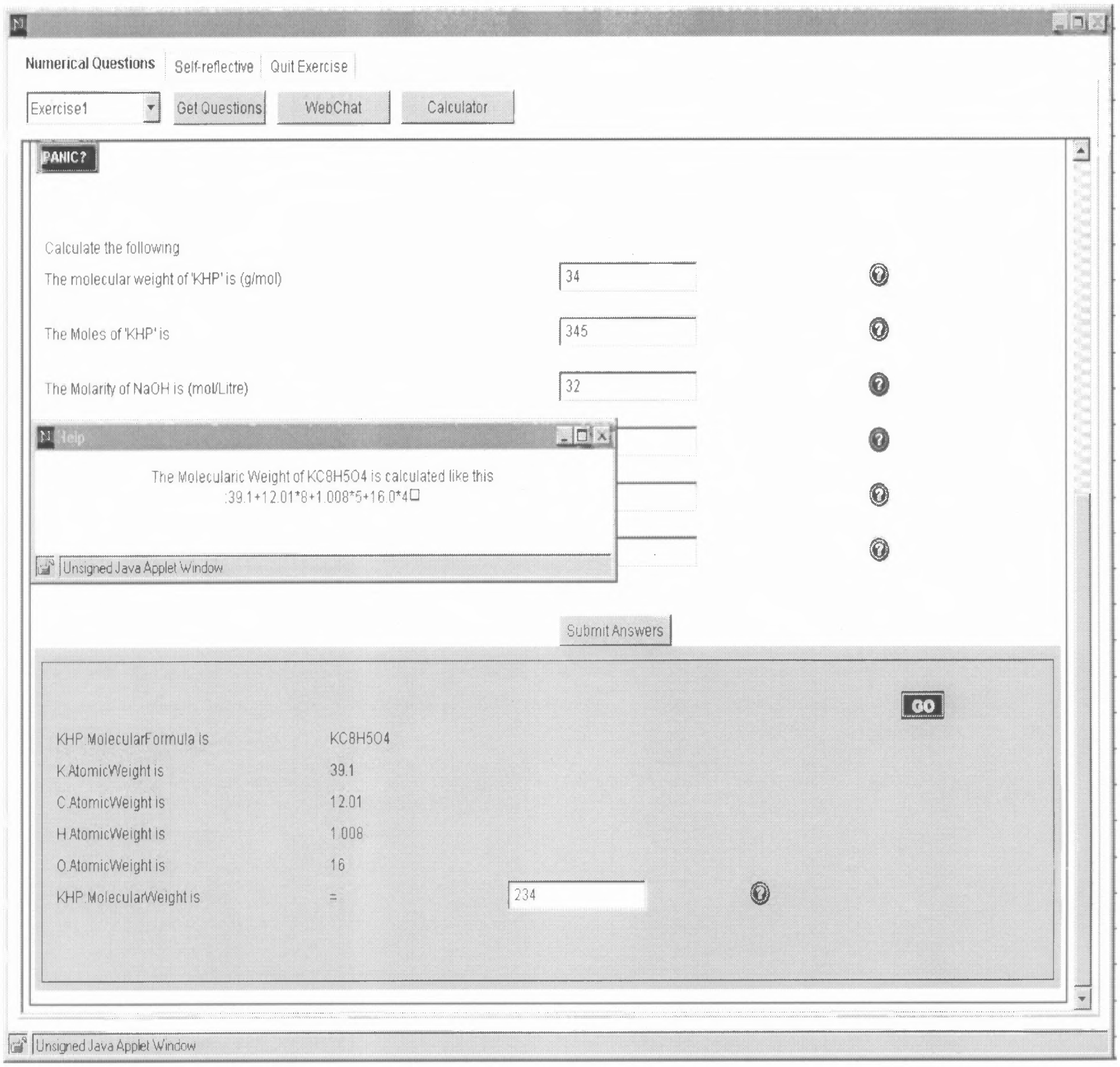
National Science Foundation - Division of Undergraduate Education-CCP DUE-9752483

Figure 7.1 The General Chemistry Courseware Homepage

7.1.2 DISTILS Prototype Implementation

The general chemistry courseware presently consists of twelve laboratory experiments. The DISTILS prototype enhances three of these, namely: Measuring Density, Titration of Acidic Substances and pH and Buffers.

Java is used as the implementation language and CORBA compliant software--OrbixWeb--is used as the object request broker for serving as communication infrastructure. Figures 7.2- 7.4 present scenarios of the prototype system.



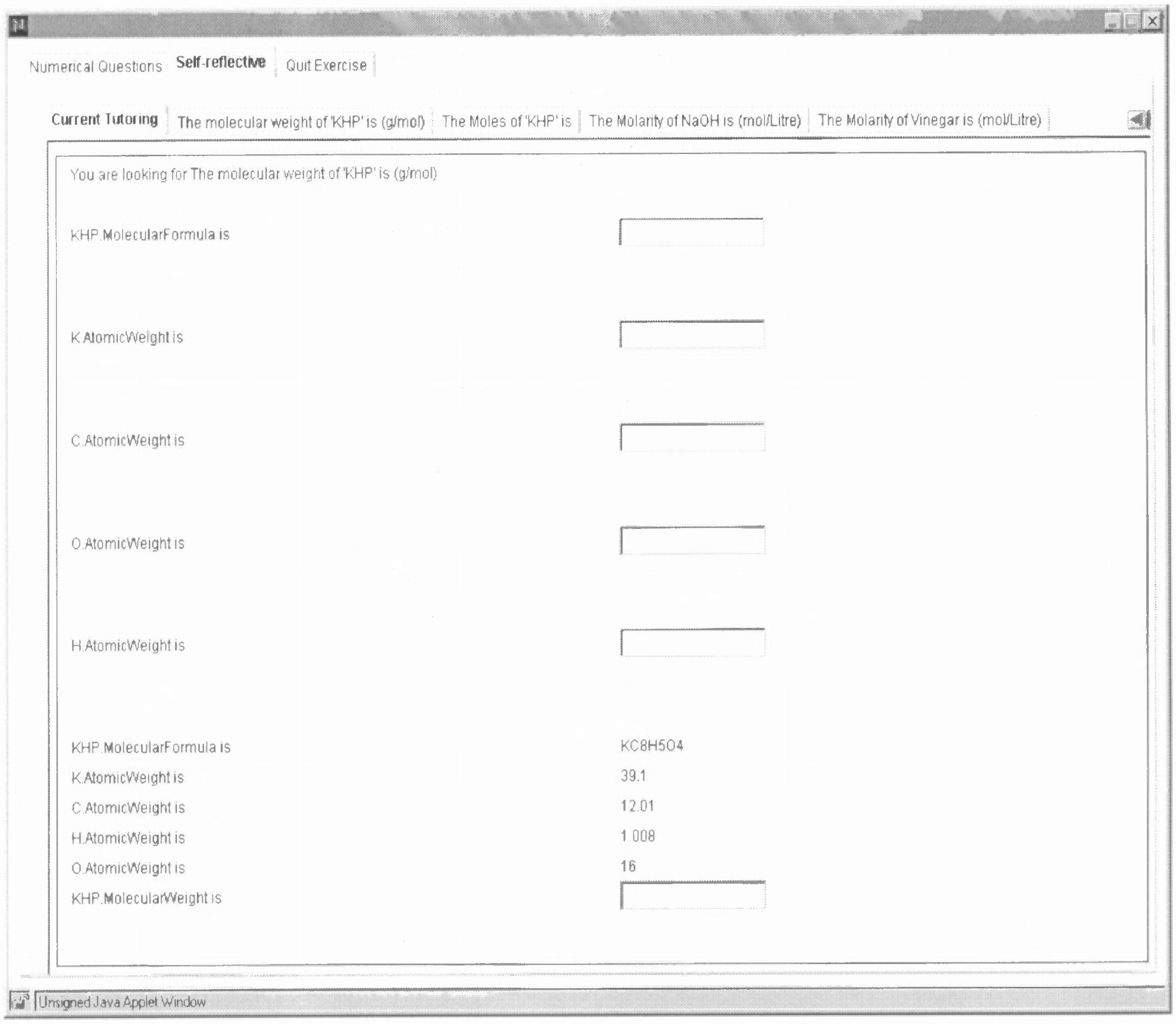
Calculate the following

The molecular weight of 'KHP' is (g/mol)	34
The Moles of 'KHP' is	345
The Molarity of NaOH is (mol/Litre)	32

The Molecularic Weight of KC8H5O4 is calculated like this
:39.1+12.01*8+1.008*5+16.0*4□

KHP MolecularFormula is	KC8H5O4	
K AtomicWeight is	39.1	
C AtomicWeight is	12.01	
H AtomicWeight is	1.008	
O AtomicWeight is	16	
KHP MolecularWeight is	=	234

Figure 7.2 An Example of Using ITT in Acid Titration Laboratory.


Numerical Questions
Self-reflective
Quit Exercise
Current Tutoring
The molecular weight of 'KHP' is (g/mol)
The Moles of 'KHP' is
The Molarity of NaOH is (mol/Litre)
The Molarity of Vinegar is (mol/Litre)
Unsigned Java Applet Window

You are looking for The molecular weight of 'KHP' is (g/mol)

KHP.MolecularFormula is

K.AtomicWeight is

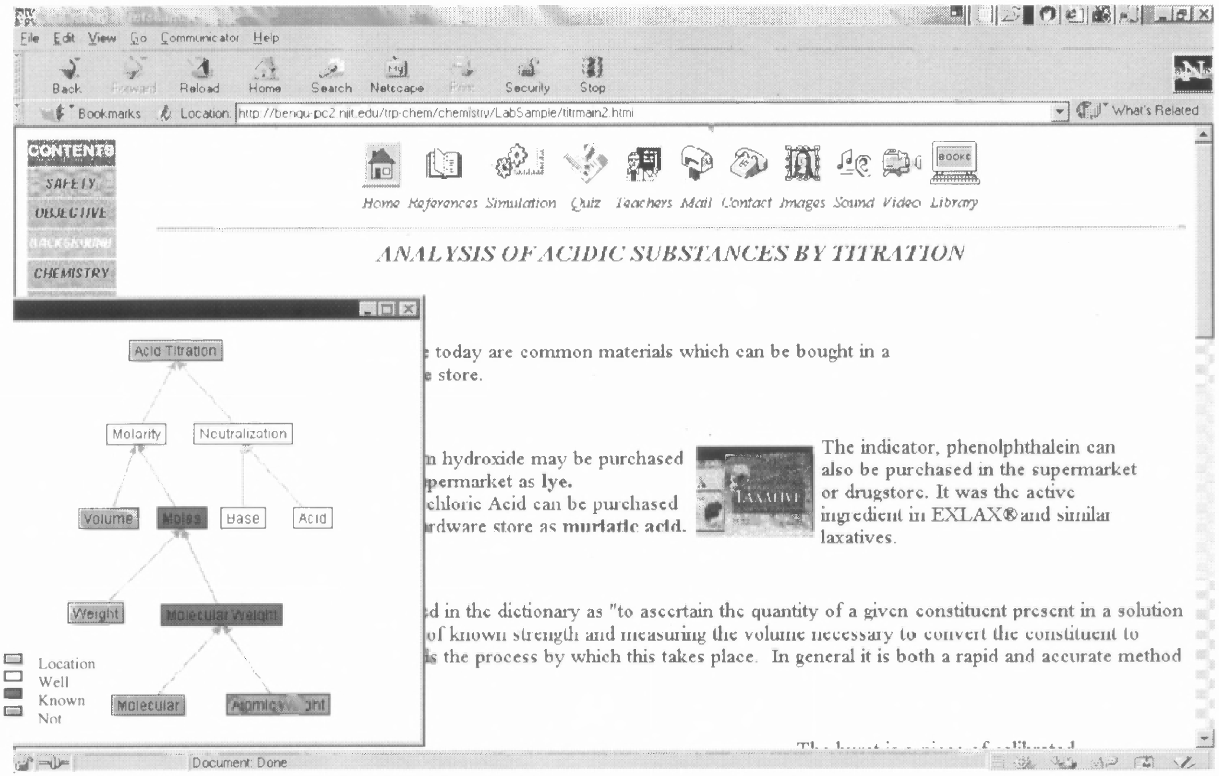
C.AtomicWeight is

O.AtomicWeight is

H.AtomicWeight is

KHP.MolecularFormula is	KC8H5O4
K.AtomicWeight is	39.1
C.AtomicWeight is	12.01
H.AtomicWeight is	1.008
O.AtomicWeight is	16
KHP.MolecularWeight is	

Figure 7.3 Student Reflects Current Tutoring Session



http://bengu-pc2.njit.edu/trp-chem/chemistry/LabSample/titrmain2.html

ANALYSIS OF ACIDIC SUBSTANCES BY TITRATION

e today are common materials which can be bought in a
e store.

n hydroxide may be purchased
permarket as **lye.**
chloric Acid can be purchased
rdware store as **muriatic acid.**

The indicator, phenolphthalein can also be purchased in the supermarket or drugstore. It was the active ingredient in EXLAX® and similar laxatives.

d in the dictionary as "to ascertain the quantity of a given constituent present in a solution
of known strength and measuring the volume necessary to convert the constituent to
is the process by which this takes place. In general it is both a rapid and accurate method

Figure 7.4 An Example of Using ALG in Acid Titration Laboratory

Students interact with the ITT through the user interface, which is developed as a Java applet and can be accessed through web browsers. Students select problems and initiate further help through the interface. When a student selects a problem, the interface components request ORB server to create a particular ITT on the particular problem. The problem is formatted and displayed for the student to solve. On the other side, the ITT begins to create problem space, during which related knowledge objects are initiated and student modeler is created. Knowledge objects begin to cooperatively solve the problem and record solution activities on the blackboard while the student is solving on the remote site. After the student submits the solution, answers are checked and marked if there are any errors. The student may then ask for help and the problem-solving coaching session starts.

Figure 7.2 shows a scenario of a student working with an ITT in the acid titration laboratory experiment. In this particular case, this student did not know how to calculate the molecular weight of the KHP and asked for help. The ITT prompted the student with all prerequisite knowledge such as the molecular formula of KHP, necessary atomic weights and so on, but the student still had difficulty in getting the answer. The ITT then demonstrated to the student how the KHP's molecular weight is calculated.

After a student finished solving an exercise, s/he can review their problem solving processes and compare it with the solution provided by the expert agents (Figure 7.3). The current tutoring panel captures the tutoring interaction between the student and the intelligent tutor tool. The other panels present how knowledge objects apply related knowledge to solve particular questions. This could help students to establish a global knowledge map for the problem, which is not sufficiently stressed during the step-by-step coaching process.

Figure 7.4 shows an example of using the ALG in the ACID TITRATION Laboratory. The student was not sure if he was well prepared for this laboratory and asked for guidance. In the diagram, Acid Titration is colored in green. Molarity, Neutralization, Base and Acid are colored in white, Moles and Molecular Weight are in red, and Volume, Weight, Molecular and Atomic Weight are in gray. The student in this situation then knows that he/she needs to go through the section on Molecular Weight first, and click on it to go to that section. The ALG agent is expected to help students who are novices in the subject matter or have less navigating proficiency.

7.2 Formative Evaluation

7.2.1 Hypothesis

It is hypothesized that the DISTILS prototype would effectively improve students' learning in a Web-based learning environment. Compared to a control group students who learned in a traditional Web-based environment without the DISTILS prototype, students who used in the same Web-based environment with DISTILS will perform better.

7.2.2 The Experiment

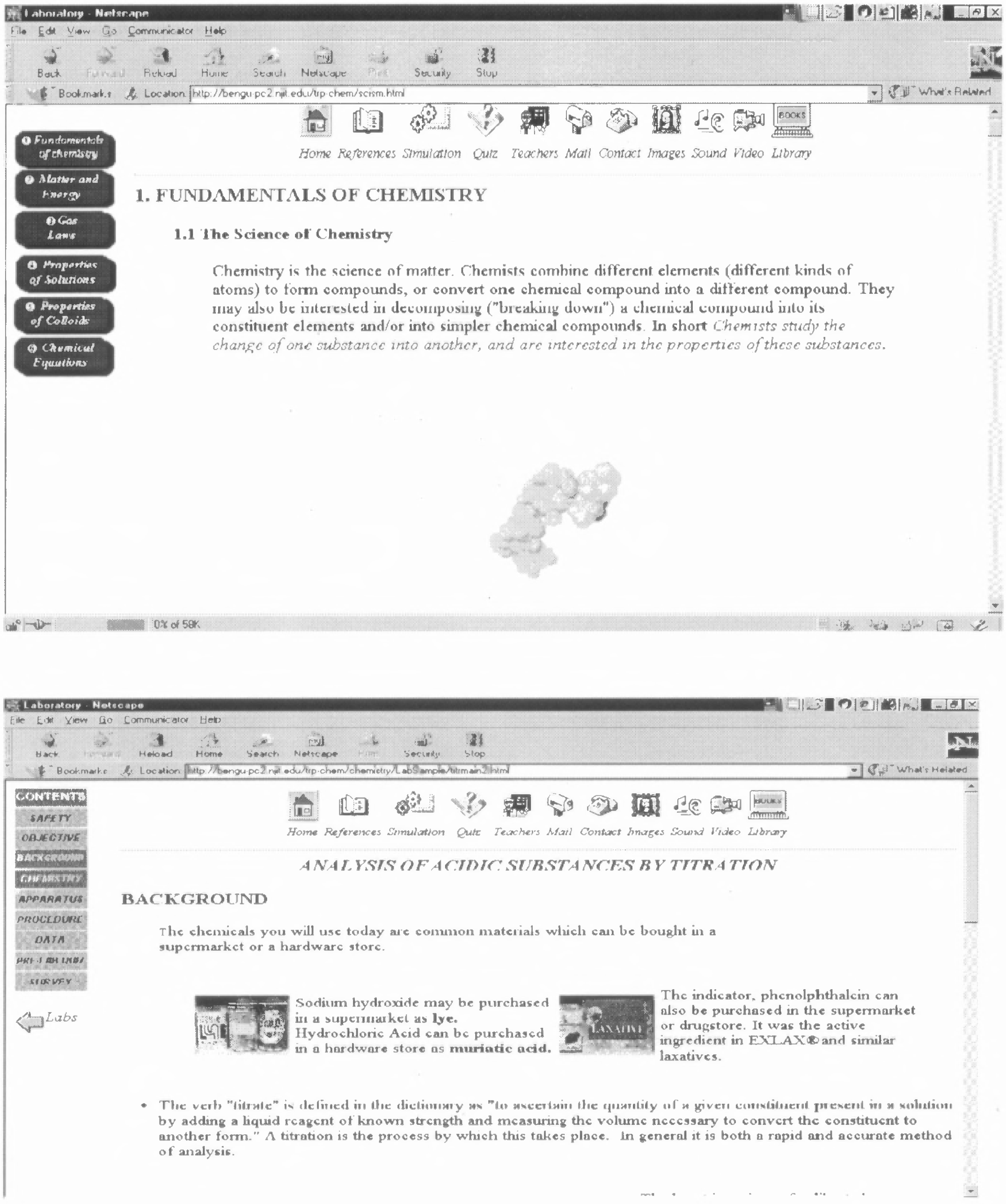
7.2.2.1 Subjects: Students with a high school level chemistry background participated in this experiment. Three high school students who participated in a summer internship at New Jersey Center for Multimedia Research (NJCMR) and 22 NJIT students participated in the evaluation. This set of students was divided into two groups. Students were randomly assigned to one of the two groups each time they logged on.

7.2.2.2 Manipulation of Independent Variables In this experiment, one independent variable- DISTILS, which is featured by the ALG and ITT, is defined. We simply present or withdraw DISTILS from the traditional Web-based learning environment of chemistry lab. The experiment objective is to compare students' performance in a traditional Web-based environment vs Web-based environment with DISTILS. In this study, a traditional Web-based learning environment is defined as a Web environment without any advanced feature developed for educational purposes. Particularly, in this experiment it refers to this developed courseware with DISTILS disabled.

7.2.2.3 Measurement of Dependent Variables: The major dependent variable is student's test performance. The student performance will be measured based on the quiz score. The number of the correct answers is used as the quiz score.

7.2.2.4 Material Preparation: The prototype includes three laboratory experiments of the on-line chemistry course: 1) Measuring the Density of a Solid and a Liquid, 2) Analysis of Acidic Substances by Titration, and 3) pH and Buffers.

Each laboratory includes structured reading material on Safety, Objective, Background, Apparatus, Procedure and Data. Students are expected to study these material and then take the pre-lab quiz. For students who do not understand the material well, a hypertext version of Science of Chemistry was provided for their further study on fundamental chemistry concepts (**http://www.njcmr.org/distils**). Figure 7.5 illustrates this material.


Laboratory - Netscape
File Edit View Go Communicator Help
Back Forward Reload Home Search Netscape Print Security Stop
Bookmarks Location: http://bengu-pc2.njit.edu/trp-chem/scism.html
What's Related
Home References Simulation Quiz Teachers Mail Contact Images Sound Video Library
Fundamentals of chemistry
Matter and Energy
Gas Laws
Properties of Solutions
Properties of Colloids
Chemical Equations
0% of 58K

1. FUNDAMENTALS OF CHEMISTRY

1.1 The Science of Chemistry

Chemistry is the science of matter. Chemists combine different elements (different kinds of atoms) to form compounds, or convert one chemical compound into a different compound. They may also be interested in decomposing ("breaking down") a chemical compound into its constituent elements and/or into simpler chemical compounds. In short *Chemists study the change of one substance into another, and are interested in the properties of these substances.*

Laboratory - Netscape
File Edit View Go Communicator Help
Back Forward Reload Home Search Netscape Print Security Stop
Bookmarks Location: http://bengu-pc2.njit.edu/trp-chem/chemistry/LabSample/titmain2.html
What's Related
Home References Simulation Quiz Teachers Mail Contact Images Sound Video Library
CONTENTS
SAFETY
OBJECTIVE
BACKGROUND
CHEMISTRY
APPARATUS
PROCEDURE
DATA
PRELIMINARY
SURVEY
Labs

ANALYSIS OF ACIDIC SUBSTANCES BY TITRATION

BACKGROUND

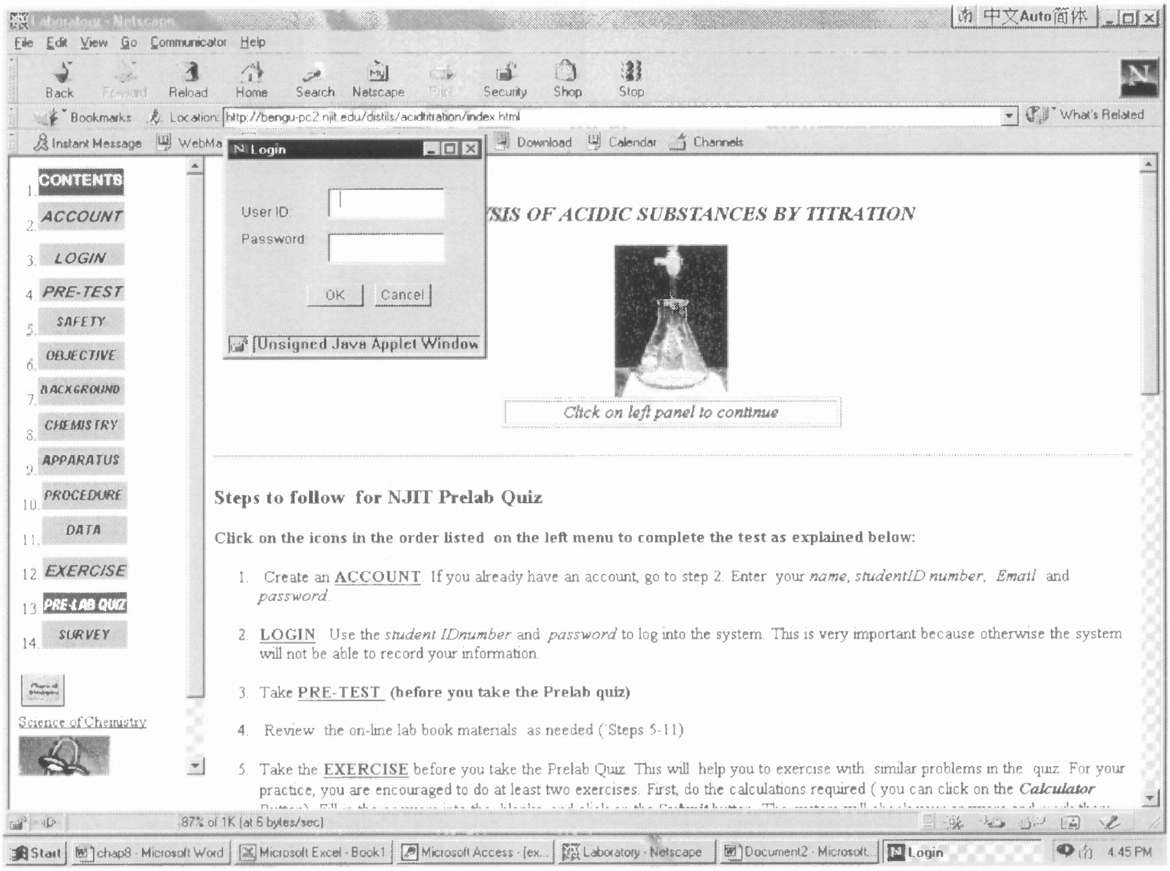
The chemicals you will use today are common materials which can be bought in a supermarket or a hardware store.

Sodium hydroxide may be purchased in a supermarket as **lye.** Hydrochloric Acid can be purchased in a hardware store as **muriatic acid.**

The indicator, phenolphthalein can also be purchased in the supermarket or drugstore. It was the active ingredient in EXLAX® and similar laxatives.

- The verb "titrate" is defined in the dictionary as "to ascertain the quantity of a given constituent present in a solution by adding a liquid reagent of known strength and measuring the volume necessary to convert the constituent to another form." A titration is the process by which this takes place. In general it is both a rapid and accurate method of analysis.

Figure 7.5 Experiment Materials: a)Science of Chemistry b)Laboratory Material


Laboratory - Netscape
File Edit View Go Communicator Help
Back Forward Reload Home Search Netscape Print Security Shop Stop
Bookmarks Location: http://bengu-pc2.njit.edu/distils/acidtitration/index.html
What's Related
Instant Message WebMail Download Calendar Channels
Login
User ID
Password
OK Cancel
Unsigned Java Applet Window
CONTENTS
1.
2. ACCOUNT
3. LOGIN
4. PRE-TEST
5. SAFETY
6. OBJECTIVE
7. BACKGROUND
8. CHEMISTRY
9. APPARATUS
10. PROCEDURE
11. DATA
12. EXERCISE
13. PRE-LAB QUIZ
14. SURVEY
Science of Chemistry
87% of 1K (at 6 bytes/sec)
Start chap8 - Microsoft Word Microsoft Excel - Book1 Microsoft Access - [ex... Laboratory - Netscape Document2 - Microsoft... Login 4:45 PM

...SIS OF ACIDIC SUBSTANCES BY TITRATION

Click on left panel to continue

Steps to follow for NJIT Prelab Quiz

Click on the icons in the order listed on the left menu to complete the test as explained below:

1. Create an ACCOUNT. If you already have an account, go to step 2. Enter your *name, studentID number, Email* and *password*.
2. LOGIN. Use the *student IDnumber* and *password* to log into the system. This is very important because otherwise the system will not be able to record your information.
3. Take PRE-TEST **(before you take the Prelab quiz)**
4. Review the on-line lab book materials as needed (Steps 5-11)
5. Take the EXERCISE before you take the Prelab Quiz. This will help you to exercise with similar problems in the quiz. For your practice, you are encouraged to do at least two exercises. First, do the calculations required (you can click on the ***Calculator***

Figure 7.6 Acid Titration Experiment System

Each laboratory also includes several exercises and a pre-lab quiz. Students are required to pass the pre-lab quiz before they perform the experiment in the laboratory. Figure 7.6 shows the experimental system for the ACID TITRATION.

7.2.2.5 Experimental Design: In this experiment, students were divided into two groups: the control-group and experimental group (Wickens et al 1998). Students were also asked to do before and after tests. The control group students tested under the traditional Web-based learning environment, the experimental group students tested under the adaptive learning environment with DISTILS prototype.

Each student was given an account to log in. After they logged in, the system kept their learning records and automatically stored these in the database for analysis. The students were randomly assigned to one of the two groups. Both groups of students first took a prior test before the experiment to assess their knowledge background. Then they were asked to navigate the lab manual and on-line chemistry textbook to learn the necessary skills, during which they can use some exercises for self-evaluation of their understanding. Finally they were asked to take a quiz and fill in a usability questionnaire.

Both groups of students read the same material and took the same prior test, data used for exercises and quiz were randomized so that the answers were different. The only difference between the two groups was that in the experimental group, the exercises were presented with ITT and ALG of DISTILS, while in the control group, exercises were not associated with the ITT, so that students who did the exercise only knew if their answers were correct but not given guidance. Thus, any learning benefits were due to integration of the DISTILS prototype. The answers, the number of trials and the time each student spends with the pre-laboratory quiz are automatically logged.

The DISTILS prototype was implemented in three laboratory experiments: Density, Titration of Acidic Substances, pH and Buffers. During the evaluation, Density experiment was used for demonstration to get students familiar with the system. All twenty-five students participated in Acid Titration experiment, and only three students participated in pH and Buffers experiments.

7.2.3 Descriptive Method

Descriptive method was used to collect students' backgrounds, responses and comments for analysis. A usability questionnaire is developed based on Schneiderman's (1998) work. This questionnaire covers questions about students' computer experience, chemistry & math experiences, their reaction to the learning tool, knowledge learned from the learning tool and their final evaluation and suggestions. The questionnaire is listed in Appendix B.

7.3 Results Analysis

In this section, the results in Acid Titration experiment are analyzed. Table 7.1 shows students' scores in both the pre-test and the final quiz. Table 7.2 shows the assessment results collected from student feedback. In this experiment, thirteen students participated in the experimental group and twelve students participated in the control group. The feedback of one student in the experimental group was missed.

Table 7.1 Scores in Pre- and Final Test (In the Acid Titration experiment, the total number of questions in Pre- andFinal test are 6 and 7, respectively. The score in the pre-test and final quiz is the number of correct answers.)

Experimental Group				Control Group			
Subject	Pre-Score	Final Score	Score Gain	Subject	Pre-score	Final Score	Score Gain
1	1	7	6	4	1	4	3
2	1	7	6	7	0	4	4
5	1	5	4	8	1	3	2
6	0	5	5	10	1	2	1
9	2	6	4	11	1	4	3
12	2	6	4	13	1	1	0
14	0	3	3	16	2	5	3
15	1	3	2	3	1	1	0
17	2	6	4	18	3	2	-1
19	0	6	6	23	2	5	3
20	0	6	6	21	0	1	1
24	0	3	3	25	0	1	1
22	0	6	6				

Table 7.2 Student Assessment Results on DISTILS (Where: OSE-Overall system evaluation, SE-System easy of use, OLM-On line material, OLE-On line exercise, HMLearn-How much you learned, Chem-How familiar with Chemistry. The feedback from subject 17 was missed and was not shown).

Subject	OSE	SE	OLM	OLE	Speed	HMLearned	Chem	Group
1	6	6	5	7	3	6	5	
2	5	4	3	7	4	4	4	
5	5	5	3	5	3	3	3	
6	6	5	4	6	4	5	5	
9	6	4	5	6	5	4	3	
12	5	5	4	5	3	4	3	Experiment
14	4	4	6	6	3	5	6	
15	3	3	5	3	1	3	6	
19	7	6	7	7	6	2	4	
20	3	4	6	6	6	1	2	
24	6	3	7	7	1	2	5	
22	7	6	7	7	7	2	5	
17								
3	3	1	4	2	1	2	3	
4	4	4	4	2	2	3		
7	2	1	3	2	2	2	4	
8	3	2	3	2	4	2	1	
10	3	2	3	1	2	2	2	
11	4	4	4	3	2	3	2	
13	4	3	5	6	3	5	5	Control
16	7	7	4	7	6	6	3	
18	5	4	5	4	6	3	3	
21	5	5	4	6	3	3	6	
23	1	3	7	6	3	4	2	
25	6	5	6	7	7	1	3	

Students' scores in both pre- and final test were examined. The *t*-test is used to test the significance of the difference between the two groups since a small sample was used (Clark 1969). The procedure of *t*-test includes three steps. The first step is to calculate a pooled variance, which is a variance based on samples. The second step is to calculate the observed *t* value. The third step is to compare the observed value with a tabulated value. If the observed value is greater than the designated tabulated value, the null hypothesis that the two samples are not significantly different, should be rejected at significance level α. Else the null hypothesis can not be rejected. Equation 7.1 and equation 7.2 is used to calculate the pooled variance and the *t* value.

$$s^2 = \frac{1}{N+M-2}\left\{\sum_{i=1}^{N}(x_i-\hat{x})^2 + \sum_{j=1}^{M}(y_j-\hat{y})^2\right\} \tag{7.1}$$

$$t_{(N+M-2)} = \frac{\hat{x}-\hat{y}}{\sqrt{s^2(\frac{1}{N}+\frac{1}{M})}} \tag{7.2}$$

$$\begin{aligned} &wher: N \ and \ M \ is \ the \ sample \ size \ of \ the \ two \ group \\ &\hat{x}, \hat{y} \ are \ the \ mean \ of \ the \ two \ group \ respectively \end{aligned}$$

Three tests were performed on the Pre- and Final Test scores: a *t*-test on the pre-test score mean difference between the experimental and control group, a *t*-test on the final quiz score mean difference between the experimental and control group, a paired *t*-test on the learning improvement between the experimental group and the control group.

In the following test, the following symbols are defined:

μ_{pe}: the mean of the pre-test score for the experimental group

μ_{fe}: the mean of the final quiz score for the experimental group

μ_{pc}: the mean of the pre-test score for the control group

μ_{fc}: the mean of the final quiz score for the control group

μ_{ge}: the mean of the pre- and final test score gain for the experimental group

μ_{gc}: the mean of the pre- and final test score gain for the control group

Test 1:

H0: $\mu_{pe} - \mu_{pc} = 0$

Ha: $\mu_{pe} - \mu_{pc} \neq 0$

Table 7.3 shows the *t*-test results on the pre-test score mean difference between the experimental and control group. In this test, N=13, M=12, α=0.05. The two tail *t* critical value is 2.07 and the observed *t* statistic value is -0.91. Since $-2.07 < -0.91 < 2.07$, the hypothesis H0 is accepted. Therefore, there is no statistical difference in the two group's score in the pre-test test.

Test 2:

H0: $\mu_{fe} - \mu_{fc} = 0$

Ha: $\mu_{fe} - \mu_{fc} > 0$

Table 7.4 shows the *t*-test results on the final quiz score mean difference between the experimental and control group. In this test, N=13, M=12, α=0.05. The upper tail *t* critical value is 1.71 and the observed *t* statistic value is 4.21. Since $1.71 < 4.21$, the hypothesis H0 is rejected and the alternative hypothesis Ha is accepted. Therefore, the experimental group's score in the final test was greater than the control group's score in the final test, and this difference was statically significant (p=0.00017) at α=0.05 levels of significance.

Test 3:

H0: $\mu_{ge} - \mu_{gc} = 0$

Ha: $\mu_{ge} - \mu_{gc} > 0$

Table 7.5 shows the paired *t*-test results on the before- and after- test score gains between the experimental group and the control group. In this test, $\alpha=0.05$. The upper tail *t* critical value is 1.78 and the observed *t* statistic value is 4.87. Since 4.87 > 1.78, the hypothesis H0 is rejected and the alternative hypothesis Ha is accepted. Therefore, the experimental group's learning improvement during this experiment was greater than the control group's (p=0.00003).

Test 1 showed that there was no significant knowledge background difference between the control group students and the experimental group students. Test 2 showed that students in the experimental group scored significant better than those in the control group. Test 3 showed that students in the experimental group gained more significant improvement than those in the control group did.

Table 7.3 t-Test: Pre-Test Score of The Experimental and Control Group Assuming Equal Variances

	Experimental Group	Control Group
Mean	0.77	1.08
Variance	0.69	0.81
Observations	13	12
Pooled Variance	0.75	
Hypothesized Mean Difference	0	
df	23	
t Stat	-0.91	
P(T<=t) two-tail	0.37	
t Critical two-tail	2.07	

Table 7.4 t-Test: Final Quiz Score of The Experimental and Control Group Assuming Equal Variances

	Experiment Group	Control Group
Mean	5.3	2.75
Variance	2.06	2.57
Observations	13	12
Pooled Variance	2.31	
Hypothesized Mean Difference	0	
df	23	
t Stat	4.21	
P(T<=t) one-tail	0.00017	
t Critical one-tail	1.71	

Table 7.5 t-Test: Learning Improvement of The Experimental and Control Group Assuming Equal Variances

	Experimental Group	Control Group
Mean	4.54	1.67
Variance	1.94	2.42
Observations	13	12
Pooled Variance	2.17	
Hypothesized Mean Difference	0	
df	23	
T Stat	4.87	
P(T<=t) one-tail	3.21E-05	
T Critical one-tail	1.71	

Table 7.6 shows the mean of student evaluation on survey items in the experiment. On the overall system evaluation, students in the experimental group evaluated 5.25 while students in the control group gave 3.9. On ease of use, the experiment group students evaluated the system at 4.58 while the control group students

evaluated at 3.42. It was also found that exercises with the ITT were more favorable than exercise without the ITT (the experiment group students evaluated the on line exercise 6 while the control group students evaluated 4). Both group students agreed that the system speed needs to be improved (the evaluation of the experiment group and the control group on System Speed is 3.8, 3.4, respectively). It was also found that there was only small difference when both group students were asked the question" How much you learned?". The experiment group students evaluated 3.4 while the control group students evaluated 3. A possible explanation for it was that the time period for this experiment is short. It was concluded from the above results that at average level students found that the Web-based environment with the DISTILS prototype more favorable than the one without DISTILS.

Table 7.6 Results of Student Evaluation (Where: OSE-Overall system evaluation, SE-System easy of use, OLM-On line material, OLE-On line exercise, HMLearn-How much you learned, Chem-How familiar with Chemistry.)

Group		OSE	SE	OLM	OLE	Speed	HMLearn	Chem
Experiment	Mean	5.25	4.58	5.17	6	3.83	3.42	4.25
	StdDev	1.36	1.08	1.47	1.21	1.90	1.51	1.29
Control	Mean	3.92	3.42	4.33	4	3.42	3	3.09
	StdDev	1.68	1.78	1.23	2.26	1.93	1.41	1.45

CHAPTER 8

CONCLUSIONS AND FUTURE WORK

8.1 Conclusions

In general, this dissertation makes two important contributions: 1) the mechanism for Web-based adaptive learning and its effectiveness in general chemistry laboratory education domain is investigated. 2) Distributed object technology --CORBA-- is for the first time introduced into the design of intelligent learning applications. A generic CORBA-based object-oriented framework is developed for the cooperative development of reusable intelligent educational objects, and has the potential to improve the cost-effectiveness of development of intelligent educational materials.

8.1.1 The Mechanism and Impact of Web-based Adaptive Learning

A Web-based adaptive learning paradigm that consists of adaptive navigation guidance and an on-line intelligent tutoring tool is developed. Adaptive navigation guidance prevents students from being disoriented and reduces their cognitive load when a unfamiliar domain is presented in Web format. The on-line intelligent tutoring tool provides students timely problem solving support in a dynamic Web environment. The prototype, DISTILS, has been implemented in a general chemistry laboratory education domain.

Preliminary results showed that DISTILS effectively enhanced learning in Web environment. Three high school students and twenty-two NJIT students participated in the evaluation of DISTILS. In the final quiz of 7 questions, the average correct answers of the students who studied in a Web environment with DISTILS (DISTILS Group) was 5.3, and the average correct answers of the students who studied in the same Web environment without DISTILS (NoDISTILS Group) was 2.75. A t-test conducted on this small sample showed that the DISTILS group students scored significantly better than the NoDISTILS group students.

8.1.2 A Generic CORBA-driven Framework for Development of Reusable Intelligent Educational Objects

A generic CORBA-driven framework is used for the development of reusable intelligent educational objects. In this framework, knowledge-objects model domain expertise, a student modeler assesses the student's knowledge progress, an intelligent tutoring tool provides problem-solving coaching and an adaptive-lecture-guidance assists students through new domains. The CORBA-compatible middleware serves as the communication infrastructure. These objects are easily integrated in a reusable, plug-and-play manner. In the DISTILS prototype, several knowledge objects were developed and are reused in the development of different chemistry laboratory experiments. Given the platform independence capability of CORBA, these objects could be also used in other environments.

8.2 Suggestions for Future Research

This study is an initial experiment with the development of CORBA-driven reusable intelligent learning objects in Web-based environment. The prototype demonstrates the effectiveness of the approach in this dissertation. However, it is also realized that there are significant problems in the initial stage of this study. Many aspects have to be improved to fully meet our research objectives.

8.2.1 Mechanism of Explicit Knowledge in Object

Since CORBA IDL lacks an explicit knowledge description mechanism, the objects developed in DISTILS have implicitly embodied their knowledge. This is a big disadvantage of the prototype implementation. To solve this problem, a powerful symbolic expression and inference mechanism has to be developed based on CORBA.

8.2.2 Problem-solving Planning and Conflict Resolution

In a problem-solving process, the task planning and conflict resolution mechanism is important but very difficult. In DISTILS, we skipped task planning by taking advantage of carefully designed questions. We used a simple back-tracing technique to replace conflict resolution. These techniques are applicable in DISTILS because our prototype system deals with a relatively small domain. When facing larger and more complex domain, planning object and conflict resolution object need to be integrated.

8.2.3 Neural Network and Representation of Deeper Learner Model

In DISTILS, simple functions have been used to predict student knowledge performance. This technique is straightforward but limited. A human teacher is always able to sense the student's deep learning trends from previous tutoring experience and make predictions of what the student will do in a given context. Neural network has great potential in representing and discovering this kind of learner model. While neural network has difficulty with local minima during learning process, the genetic algorithm is robust for global optimization and is suitable to training these kind of neural networks. However, application of neural network in student modeling needs large valid training sample. As the DISTILS prototype gradually becomes mature and once it is reliable in a practical application, much data can be collected and neural network techniques for student modeling will be practical.

8.2.4 Standardization on Intelligent Educational Objects

This dissertation demonstrates the first effort of designing a CORBA-based object-oriented framework for reusable intelligent educational object development. However, much work needs to be done before this approach will be practical. While the IDL interface defined for learning objects is effective in the development of the prototype, we reasonably believe that it is limited and may be not applicable to other domains. The interface definition and fundamental services in intelligent educational objects need to be standardized. Since the Object Management Group the CORBA consortium, has contributed much to standardization in other domains such as finance and manufacturing, we propose that it could also make a new contribution to the intelligent educational content domain.

APPENDIX A

QUESTIONS USED IN EXPERIMENTS

This material lists the questions used in the evaluation experiments.

1. pH and Buffer

Pre-Test
Vitamin C is the common name for ascorbic acid, HC6H7O6. A solution is made by dissovling a tablet of pure vitamin C in water and diluting it.

Ka of Vitamin C is 0.00008
Volume of the solution: (ml) 105.3
Mass of the tablet weighed out (grams) 0.713

Calculate the following
1) What is the molarity of the solution null
2) What is the pH of this solution null

Exercise 1
A weak monoprotic acid is weighted out, and diluted to 50.00ml.
Exactly half of this is titrated with
NaOH until a pale pink phenolphthalein end point is reached. This titrated solution is mixed with the unititrated
half of the original solution and the pH of the mixture is measued.

Mass of the acid (g) 0.698
pH of original acid solution 2.98
pH of half-titrated mixture 4.75

Calculate the following
1) What is the pKa of the acid? null
2) What is the molarity of the original acid solution null
3) What is the molecular weight of the acid? null

Exercise 2
Vitamin C is the common name for ascorbic acid, HC6H7O6. A solution is made by dissovling
a tablet of pure vitamin C in water and diluting it.

Ka of Vitamin C is 0.00008
Volume of the solution: (ml) 110.7
Mass of the tablet weighed out (grams) 0.664

Calculate the following
1) What is the molarity of the solution null

2) What is the pH of this solution null

Pre-Lab Quiz
A solid weak acid is weighed, dissolved in water and diluted to exactly 50.00 ml. 25.00
ml of the solution is taken out and is titrated to a neutral endpoint with 0.10 M NaOH.
The titrated portion is then mixed with the remaining untitrated portion and the pH of the
mixture is measured.

Mass of acid weighed out (grams) 0.633
Volume of NaOH required to reach endpoint: (ml) 19.1
pH of the mixture (half neutralized solution) 3.07

Calculate the following
1) What is the Pka of the acid? null
2) What is the molarity of the original acid solution null
3) What is the molecular weight of the acid? null
4)If 0.001 moles of HCl is added to the half neutralized solution, what will the new pH be? null

2. Density

Pre-Test
Determining the density of a solid
1) You weigh a dry flask and a chunk of solid.
2) You then place the solid into the flsk and fill the flask to the top with water and weigh the full flask
3),you take the solid out,and fill the flask to the top with water and weigh it once more.
Density water at 20 C = .9982 Density water at 35 C = .9941

Mass of flask in grams 58.99
Mass of solid in grams 90.97
Mass of flask and water (g) 83.2543
Mass of flask and solid and water (g) 165.1089
Measured temperature of water (C) 22.9546

Calculate the following
The density of water at the given temperature is (g/cc) null
The volume of the flask is (cc) null
The volume of the solid is (cc) null
Density of the solid unknown is (g/cc) null

Exercise
Determining the density of a solid
1) You weigh a dry flask and a chunk of solid.
2) You then place the solid into the flsk and fill the flask to the top with water and weigh the full flask
3),you take the solid out,and fill the flask to the top with water and weigh it once more.
Density water at 20 C = .9982 Density water at 35 C = .9941

Mass of flask in grams	58.99
Mass of solid in grams	98.93
Mass of flask and water (g)	83.2543
Mass of flask and solid and water (g)	171.2339
Measured temperature of water (C)	23.1510

Calculate the following

The density of water at the given temperature is (g/cc)	null
The volume of the flask is (cc)	null
The volume of the solid is (cc)	null
Density of the solid unknown is (g/cc)	null

Pre-Lab Quiz
Determining the density of a solid
1) You weigh a dry flask, place a chunk of solid into it, and weigh it again.
2) You then fill the flask to the top with water and weigh the full flask
Finally,you take the solid out,and fill the flask to the top with water and weigh it once more.
Density water at 20 C = .9982 g/cc Density water at 35 C = .9941 g/cc

Mass of flask in grams	58.99
Mass of flask and solid (g)	108.8738
Mass of flask and solid and water (g)	118.8707
Mass of flask and water (g)	83.2640
Measured temperature of water (C)	21.6956

Calculate the following

The density of water at the given temperature is (g/cc)	null
The volume of the flask is (cc)	null
The Mass of the solid is (g)	null
The volume of the solid is (cc)	null
Density of the solid unknown is (g/cc)	null

3. Acid Titration

Pre-Test
1) You standardize some NaOH solution using HCl.
2) You then pipet out a formic acid (HCOOH) unknown and titrate it with the standardized NaOH
3) Finally you weigh out a Vitamin tablet containing a monoprotic acid (HC6H7O6) .
4) You dissolve it and titrate it with NaOH of 1).

The molarity of the HCl is (mol/Litre) 0.44
The volume of the HCl used is (ml) 12.6
Volume of NaOH to neutralize HCl in A) (ml) 14.7
Volume of formic acid titrated (ml) 9.1
Volume of NaOH needed to neutralize formic acid (ml) 13.2
Volume of NaOH needed to neutralize Vitamin C Tablet (ml) 13.4

Calculate the following
The MOLES of HCl used is null
The MOLARITY of the NaOH is (mol/Litre) null
The MOLARITY of the formic acid is (mol/Litre) null
The Molecular Weight of the formic acid is null
The WEIGHT/VOLUME % formic acid is (Kg/Litre %) null
The Molecular Weight of the PURE Vitamin C is null
The mg of ACID in the Vitamin C tablet is (mg) null

Exercise 1
You titrate some HCl solution unknown with standardized NaOH solution.

The molarity of the standardized NaOH (mol/Litre) is 0.19
The Volume of the standardized NaOH (ml) 57.9
The volume of HCl used is (ml) 10.7

Calculate the following
The molarity of the HCl used is (mol/Litre) null
The Weight/Volume% of the HCl used (kg/Litre%) null

Exercise 2
1) You standardize some NaOH solution using HCl.
2) You then pipet out a formic acid (HCOOH) unknown and titrate it with the standardized NaOH
3) Finally you weigh out a Vitamin tablet containing a monoprotic acid (HC6H7O6) .
4) You dissolve it and titrate it with NaOH of 1).

The molarity of the HCl is (mol/Litre) 0.43

The volume of the HCl used is (ml) 17.4
Volume of NaOH to neutralize HCl in A) (ml) 11.0
Volume of formic acid titrated (ml) 6.1
Volume of NaOH needed to neutralize formic acid (ml) 16.4
Volume of NaOH needed to neutralize Vitamin C Tablet (ml) 10.1

Calculate the following
The MOLES of HCl used is null
The MOLARITY of the NaOH is (mol/Litre) null
The MOLARITY of the formic acid is (mol/Litre) null
The Molecular Weight of the formic acid is null
The WEIGHT/VOLUME % formic acid is (Kg/Litre %) null
The Molecular Weight of the PURE Vitamin C is null
The mg of ACID in the Vitamin C tablet is (mg) null

Exercise 3
A NaOH solution is standardized using a sample vinegar solution

The concentration of acetic acid (CH_3COOH) in Vinegar is (g/Litre)
40.3
Volume of Vinegar needed to neutralize NaOH solution (ml) 23.4
Volume of NaOH used (ml) 25.3

Calculate the following
The Molarity of the NaOH is (mol/Litre) null

Exercise 4
A research chemist isolates a sample of nicotinic acid ($HC_6H_4NO_2$), To determine it purity,
she titrates some of the sample with standardized NaOH solution.
The reaction is : $HC_6H_4NO_2 + OH^- \rightarrow H_2O + C_6H_4NO_2^-$

The mass of the sample used (g) 0.500
The molarity of standardized NaOH is (mol/Litre) 0.13
The volume of the standardized NaOH (ml) is 43.6

Calculate the following
The mass of Nicotinic acid in the sample (g) is null

Pre-Lab Quiz
1) You standardize some NaOH solution using potassium hydrogen phthalate.
2) You then pipet out a vinegar unknown and titrate it with the standardized NaOH
3) Finally you weigh out a tablet containing a monoprotic acid .

4) You dissolve it and titrate it with NaOH of 1).

The Mass of "KHP" weighed out (g) 0.71682
Volume of NaOH to neutralize "KHP" in A) (ml) 12.4
Volume of Vinegar titrated (ml) 3.8
Volume of NaOH needed to neutralize Vinegar (ml) 16.0
MW of monoprotic acid in TABLET (g/mol) 145.399
Volume of NaOH needed to neutralize Tablet (ml) 8.6

Calculate the following
The mw of "KHP" is (g/mol) null
The MOLES of "KHP" used is null
The MOLARITY of the NaOH is (mol/Litre) null
The MOLARITY of the VINEGAR is (mol/Litre) null
The WEIGHT/VOLUME %VINEGAR is (Kg/Litre %) null
The mg of ACID in the tablet is (mg) null

APPENDIX B

USABILITY QUESTIONNAIRE

This material lists the questionnaire used in the evaluation experiments.

Personal Information

Name: ID# Age:

Part 1:User Reaction To Web-Based Pre-Lab

Overall reactions to the system:

1.1 terrible wonderful

1 2 3 4 5 6 7

1.2 frustrating satisfying

1 2 3 4 5 6 7

1.3 dull stimulating

1 2 3 4 5 6 7

1.4 difficult easy

1 2 3 4 5 6 7

1.5 Educational Value of on-line material

low high

1 2 3 4 5 6 7

1.6 Educational value of on-line exercise

low high

1 2 3 4 5 6 7

1.7 Educational value of the whole system

low high

1 2 3 4 5 6 7

1.8 Learning to operate the system

difficult easy

1 2 3 4 5 6 7

1.9 Getting started

difficult easy

1 2 3 4 5 6 7

1.10 System speed

too slow fast enough

1 2 3 4 5 6 7

Part 2. Computer Experience

2.1 Have you taken any computer courses before? ◊ Yes ◊ No

If yes, list all the computer courses you have taken before:

2.2 On the average, how much time do you spend per week on computers?

◊ Less than one hour ◊ 4 to less than 10 hours

◊ One to less than 4 hours ◊ over 10 hours

On the scale, please circle the number that most appropriately reflects your judgment.

2.3 How confident are you in using a computer?

very confident not at all

1 2 3 4 5 6 7

2.4 How much do you enjoy using computers?

very confident not at all

1 2 3 4 5 6 7

Part 3:Chemistry Experience

3.1 Have you taken any chemistry courses before? (including high school, summer programs, etc.)

◊Yes ◊No

If yes, list all the chemistry courses you have taken.

On the scale, please circle the number that most appropriately reflects your judgement.

3.2 How familiar are you with the concepts of chemistry?

very confident not at all

1 2 3 4 5 6 7

3.3 How confident are you in the practical use of these concepts in the laboratory?

very confident not at all

1 2 3 4 5 6 7

3.4 How confident are you with using these concepts in your homework assignments?

very confident not at all

1 2 3 4 5 6 7

Part 4: Knowledge

4.1 How much have you learned from this Pre-Lab about the specific chapter?

very much none

1 2 3 4 5 6 7

4.3 How much have you learned about the theory involved?

very much none

1 2 3 4 5 6 7

4.4 How confident are you in what you've learned?

very not at all

1 2 3 4 5 6 7

Part 5: Evaluation

5.1 How much did you enjoy this learning environment?

5.2 How would you evaluate this program?

5.3 Name three things that you liked best.

1.
2.
3.

5.4 Name three things that you did not like

1.
2.
3.

5.5 How would you improve it?

General Comments:

APPENDIX C

THE INFLUENCE OF DIFFERENT β VALUE

The tables in the appendix show more simulation results on the influence of difference β value on student performance prediction.

KU	DL	Ti(hrs)	Ar	Sp(β=0.08)
PS.Moles	1	24	1	
PS.Moles	1	24	1	
PS.Moles	1	2	1	
PS.Moles	1	0		0.77

KU	DL	Ti(hrs)	Ar	Sp(β=0.08)
PS.Moles	1	24	0	
PS.Moles	1	24	1	
PS.Moles	1	2	1	
PS.Moles	1	0		0.72

KU	DL	Ti(hrs)	Ar	Sp(β=0.05)
PS.Moles	1	24	1	
PS.Moles	1	24	1	
PS.Moles	1	2	1	
PS.Moles	1	0		0.82

KU	DL	Ti(hrs)	Ar	Sp(β=0.05)
PS.Moles	1	24	0	
PS.Moles	1	24	1	
PS.Moles	1	2	1	
PS.Moles	1	0		0.73

KU	DL	Ti(hrs)	Ar	Sp(β=0.02)
PS.Moles	1	24	1	
PS.Moles	1	24	1	
PS.Moles	1	2	1	
PS.Moles	1	0		0.90

KU	DL	Ti(hrs)	Ar	Sp(β=0.02)
PS.Moles	1	24	0	
PS.Moles	1	24	1	
PS.Moles	1	2	1	
PS.Moles	1	0		0.83

KU	DL	Ti(hrs)	Ar	Sp(β=0.01)
PS.Moles	1	24	1	
PS.Moles	1	24	1	
PS.Moles	1	2	1	
PS.Moles	1	0		0.92

KU	DL	Ti(hrs)	Ar	Sp(β=0.01)
PS.Moles	1	24	0	
PS.Moles	1	24	1	
PS.Moles	1	2	1	
PS.Moles	1	0		0.85

KU	DL	Ti(hrs)	Ar	Sp(β=0.008)
PS.Moles	1	24	1	
PS.Moles	1	24	1	
PS.Moles	1	2	1	
PS.Moles	1	0		0.9333

KU	DL	Ti(hrs)	Ar	Sp(β=0.008)
PS.Moles	1	24	0	
PS.Moles	1	24	1	
PS.Moles	1	2	1	
PS.Moles	1	0		0.86

KU	DL	Ti(hrs)	Ar	Sp(β=0.08)
PS.Moles	1	24	-1	
PS.Moles	1	24	-1	
PS.Moles	1	2	-1	
PS.Moles	1	0		0.24

KU	DL	Ti(hrs)	Ar	Sp(β=0.05)
PS.Moles	1	24	-1	
PS.Moles	1	24	-1	
PS.Moles	1	2	-1	
PS.Moles	1	0		0.18

KU	DL	Ti(hrs)	Ar	Sp(β=0.02)
PS.Moles	1	24	-1	
PS.Moles	1	24	-1	
PS.Moles	1	2	-1	
PS.Moles	1	0		0.098

KU	DL	Ti(hrs)	Ar	Sp(β=0.01)
PS.Moles	1	24	-1	
PS.Moles	1	24	-1	
PS.Moles	1	2	-1	
PS.Moles	1	0		0.072

KU	DL	Ti(hrs)	Ar	Sp(β=0.008)
PS.Moles	1	24	-1	
PS.Moles	1	24	-1	
PS.Moles	1	2	-1	
PS.Moles	1	0		0.067

REFERENCES

Adeli H., Hung S. L. 1990. "Object oriented Model for Processing Earthquake Engineering Knowledge". *Microcomputers in Civil Engineering.* 5(2). 95-109.

Akagi S. 1990. "Building an Expert System for Engineering Design Based on the Object-oriented Knowledge Representation". *Journal of Mechanisms, Transmissions, and Automation in Design.* 112(2). 215-222.

Anderson J. R., Boyle C. F., Reiser B. J. 1985a. "Intelligent Tutoring System". *Science.* 228: 456-462.

Anderson J. R., Reiser B. J. 1985b. "The Lisp Tutor". *Byte.* 10(4): 159-175.

Anderson J. R., Lebiere C. 1998. *The Atomic Components of Thought.* Mahwah, NJ: Lawrence Erlbaum Associates.

Ahsani Mandana. 1998. *Literature Review and Cognitive Evaluation and Assessment of On-line Chemistry Courseware.* Internal Review at New Jersey Center for Multimedia Research (NJCMR).

Baker S. 1997. *CORBA Distributed Objects Using Orbix.* Addison-Wesley.

Bench-capon T. J. M. 1990. *Knowledge Representation: An Approach to Artificial Intelligence.* San Diego: Academic Press.

Bengu, G. 1995. "Interactive Multimedia Courseware on Manufacturing Processes & Systems". *International Journal of Engineering Education.* 11(1): 46-57.

Bengu, G., Swart W. 1996. "A Computer-Aided Total Quality Approach to Manufacturing Education in Engineering". *IEEE Transactions on Education*, 39: 415-422.

Booch G. 1994. *Object-Oriented Analysis and Design With Applications.* Benjamin/Cummings.

Brooks, D. W. 1997. *Web-Teaching: A Guide to Designing Interactive Teaching for the World Wide Web.* New York: Plenum Press.

Brusilovsky P., Schwarz E., Weber G. 1996. "ELM-ART: An Intelligent Tutoring System on World Wide Web". In Frasson C., Gauthier G. and Lesgold A.(Eds). *Intelligent Tutoring Systems, Lecture Notes in Computer Science 1086.* New York: Springer. 261-269.

Burton, R. R, Brown J. S. 1982. "An Investigation of computer coaching for informal learning activities". In Sleeman, D. H. and Brown, J. S. (Eds.). *Intelligent Tutoring Systems.* London: Academic Press.

Cao L., Bengu G. 2000. "Web-based Agents for Reengineering Engineering Education". *Journal of Educational Computing Research.* 23(4).

Carbonell J. R. 1970. "AI in CAI: an Artificial Intelligence Approach to Computer-assisted Instruction". *IEEE Trans. On Man-Machine Systems.* 11(4): 170-202.

Chappell A. R., Crowther E. G, Mitchell C. M., Govindaraj T. 1997. "The VNAV Tutor: Addressing a Mode Awareness Difficulty for Pilots of Glass Cockpit Aircraft". *IEEE Trans. on SMC, Part A.* 27(3): 327-385.

Chien S. Y. P., Xue L. Q. 1997. "Task Planning for Mobile Robot in an Indoor Environment Using Object-Oriented Domain Information". *IEEE Trans. On SMC-Part B: Cybernetics.* 27(6): 1007-1016.

Chu R. W., Mitchell C. M., Jone P. M. 1995. "Using the Operator Function Model and OFMspert as the Basis for an Intelligent Tutoring System: Towards the Tutor/Aid Paradigm for Operators of Supervisory Control Systems". *IEEE Trans. on SMC.* 25(7): 1054-1075.

Clancey W. J. 1987. "Methodology for Building an Intelligent Tutoring System". in Greg Kearsley(Ed.). *Artificial Intelligence and Instruction: Applications and Methods.* Addison-Wesley. 193-228.

Clark D. 1998. "Developing, Integrating, and Sharing Web-Based Resources for Material Education". *JOM.* 50 (5).

Clark G. M. 1969. *Statistics and Experimental Design.* American Elsevier Publishing. New York.

Detlefs David.1999. "If You Cross Lisp And Java, Do You Get Lava?" *Performance Computing.* 3: 25-28.

Devedzic V., Radovic D. 1999. "A Frmework for Building Intelligent Manufacturing Systems". *IEEE Trans. on SMC., Part C- Applications and Reviews.* 29(3). 402-419.

Eidgahy S. Y. 1998. "Engineering and the Global Marketplace: Educating "Technicians" or Problem Solvers". *ASEE Annual Comference Proceedings and CD-ROM Instructions.*

Emelyanov V. V., Iassinovski S. I. 1997. "AI-based Object-oriented Tool for Discrete Manufacturing Systems Simulation". *Journal of Intelligent Manufacturing*. 8:49-58.

Forbus Kenneth D. 1998. "Distributed Coaching for an Intelligent Learning Environment". *Proceedings of QR98.*

Fuji T. 1996. "A Case Based Approach to Collaborative Learning for Systems Analysts Education". In Frasson C., Gauthier G. and Lesgold A. (Eds.). *Intelligent Tutoring Systems, Lecture Notes in Computer Science 1086.* New York: Springer.

Gennari J. H., Cheng H., Altman R. B., Musen M. 1998. "Reuse, CORBA, and Knowledge-based Systems". *Int. J. Human-Computer Studies.* 49: 547-575.

Goldstein I. P. 1982. "The Genetic Graph: a Representation for the Evolution of Procedural Knowledge". In Sleeman, D. H. and Brown, J. S. (Eds.). *Intelligent Tutoring Systems.* London: Academic Press.

Gorti S. R., Gupta A., Kim G. J., Sriram R. D., Wong A. 1998. "An Object-oriented Representation for Product and Design Processes". *Computer-Aided Design.* 30(7): 489-501.

Gray S. H. 1993. *Hypertext and The Technology of Conversation.* Greenwood Press. Westport, CT.

Hakman M., Groth T. 1999. "Object-oriented Biomedical System Modeling-The Rationale". *Computer Methods and Programs in Biomedicine.* 59(1): 1-17.

Hoffiner Y., Facciorusso C., Field S., Schade A. 2000. "Distribution Issues in The Deisgn and Implementation of a Virtual Market Place". *Computer Networks.* 32: 717-730.

Hoque, R. 1998. *Corba 3,* IDG Books Worldwide.

IONA Technologies PLC. 1998. *OrbixWeb Programmer's Guide.*

Jacobson I. 1993. *Object-oriented Software Engineering: A Use Case Driven Approach.* Addison-Wesley.

Jagannathan V., Dodhiawala R., Baum L. S. 1989. *Blackboard Architectures and Applications.* San Diego: Academic Press.

Jerinic L, Devedzic V. 2000. "The Friendly Intelligent Tutoring Environment: Teacher's Approach". *SIGCHI Bulletin.* 32(1): 83-94.

Karacal S. C., Mize J. H. 1998. "A Formal Structure for Discrete Event Simulation. Part II: Object-oriented Software Implementation". *IIE Transactions*. 30(3): 217-226.

Kearsley G. 1993. "Cognitive Theories and Learning", Retrieved June 25, 2000 from the World Wide Web: http://www.gwu.edu/~tip

Kluiber R. W. 1996. *General Chemistry Laboratory Experiments*. Rutgers-Newark.

Kortemeyer G. 1998. "Multimedia Collaborative Content Creation-the MSU LectureOnline System". Retrieved June 25, 1998 from the World Wide Web: http://www.eoe.org/.

Koschman T., Kelson A. 1996. "Computer-Supported Problem Based Learning: A Principled Approach To the Use of Computers in Collaborative Learning". In KoschMann T.(Eds). *CSCL: Theory and Practice of an Emerging Paradigm*. New Jersey: Erlbaum.

Kozma R. B. 1991. "Learning with Media". *Review of Educational Research*. 61(2):179-211.

Kurumbalapitiya D., Ratnajcevan S. 1993. "Object-oriented Representation of Electromagnetic Knowledge". *IEEE Trans. on Magnetics*. 29(2): 1939-1942.

Lee S., O'Keefe R. M. 1996. "Effect of Knowledge Representation Schemes on Maintainability of Knowledge-based Systems". *IEEE Trans. on Knowledge and Data Engineering*. 8(1): 173-178.

Lefancois P., Montreuil B. 1994. "Object oriented Knowledge Representation for Intelligent Control of Manufacturing Workstations". *IIE Trans.* 16(1): 11-26.

Lesgold A., Lajoie S., Bunzo M., Eggan G. 1992. "SHERLOCK: A Coached Practice Environment for an Electronics Trouble Shooting Job". In Larkin J.H, Chabay R.W. (Eds.). *Computer Assisted Instruction and Intelligent Tutoring Systems*. LEA.

Lester J.C. 1996. "Focusing Problem Solving in Design-Centered Learning Environments". In Frasson C., Gauthier G. and Lesgold A. (Eds.). *Intelligent Tutoring Systems, Lecture Notes in Computer Science 1086*. New York: Springer.

Li Heng 1998. "Information-Technology-Based Tools for Reengineering Construction Engineering Education". *Comput Appl. Eng. Educ.* 6: 15-21.

Marsh II George E. 1999. "AIL 601-Theories of Learning Applied to Technological Instruction, Lecture Notes". Retrieved May 25, 1999 from the World Wide Web: http://www.bamaed.ua.edu/ail601/

Matsubara Y. 1996. "Motivation System and Human Model for Intelligent Tutoring. In Frasson C., Gauthier G. and Lesgold A.(Eds). *Intelligent Tutoring Systems, Lecture Notes in Computer Science 1086.* New York: Springer.

Milne S. 1996. "Development of a Model of User Attributes and Its Implementation Within an Adaptive Tutoring System". *User Modeling and User-Adapted Interaction.* 6: 303-335.

Ming L., Yang X. 1998. "CORBA-based Agent-Driven Design for Distributed Intelligent Manufacturing Systems". *Journal of Intelligent Manufacturing.* 9(5): 457-465.

Mitrovic, A. 1996. "INSTRUCT: Modeling Students by Asking Questions". *User Modeling and User-Adapted Interaction.* 6: 273-302.

Narayanan S., Malu P. G. 1998. "A Web-based Interactive Simulation Architecture for Airbase Logistics Systems Analysis". *International Journal of Industrial Engineering.* 5(4): 324-335.

Neyer A., Wu F. F. 1990. "Object-oriented Programming for Flexible Software: Example of A Load Flow". *IEEE Trans. on Power Systems.* 5(3): 689-696.

NSF 1992, *America's Academic Future, Directorate for Education and Human Resources.*

Norman D. A, Spohrer J. C. 1996. "Learner Centered Education". *Communications of The ACM.* 39 (4): 24-27.

O'Connor P. R. 1977. *Chemistry: Experiments and Principles.* D. C. Heath.

Obradovich J. J 1996. "The Transfusion Medicine Tutor: Usin Expert Systems Technology to Teach Domain-Specific Problem Solving Skills". in In Frasson C., Gauthier G. and Lesgold A. (Eds.). *Intelligent Tutoring Systems, Lecture Notes in Computer Science 1086.* New York: Springer.

OMG. 2000. Retrieved May 25, 2000 from the World Wide Web: http://www.omg.org

Paterson K. G. 1999. "Student Perceptions of Internet-Based Learning Tools in Environment Engineering Education". *Journal of Engineering Education,* 88(3):295-304.

Piaget Jean 1977. *The Development of Thought : Equilibration of Cognitive Structures.* Viking Press.

Principe J. C., Euliano N. R., Lefebvre W. C. 2000. "Innovating Adaptive and Neural Systems Instruction With Interactive Electronic Books". *Proceedings of the IEEE.* 88(1): 81-95.

Pope A. 1997. *The CORBA Guide: Understanding the Common Object Request Broker Architecture.* Addison-Wesley.

Raphael B., Kumar B. 1997. "Object Oriented Representation of Design Cases". *Computers & Structures.* 63 (4): 663-668.

Reiser B. J, Kimberg D. Y., Lovett M. C., Ranney M. 1992. "Knowledge Representation and Explanation in GIL: An Intelligent Tutor for Programming". In Larkin J. H, Chabay R. W. (Eds.). *Computer Assisted Instruction and Intelligent Tutoring Systems: Shared Goal and Complementary Approaches.* LEA.

Rezayat M. 2000. "The Enterprise- Web Portal for Life –cycle Support". *Computer-Aided Design.* 32:85-96.

Roschelle J. 1996. "Learning by Collaborating: Convergent Conceptual Change". In KoschMann T.(Eds). *CSCL: Theory and Practice of an Emerging Paradigm.* New Jersey-Erlbaum.

Roschelle J. , Kaput J., Stroup W., Kahn T. M. 1998. "Scaleable Integration of Educational Software: Exploring the Promise of Component Architectures". *Journal of Interactive Media in Education.* 98(6):3-22.

Sack W., Soloway E. 1992. "From PROUST to CHIRON: ITS Design as Iterative Engineering: Intermediate Results are Important!". in Larkin J. H, Chabay R. W. (Eds.). *Computer Assisted Instruction and Intelligent Tutoring Systems.* LEA.

Schank R. C. 1994. "Active Learning through Multimedia". *IEEE Multimedia (*3): 69-78.

Schank R. C., Kass A. 1996. "Goal-Based Scenario for High School Students". *Communications of The ACM.* 39(4): 28-29.

Schneiderman Ben. 1998. *Designing the User Interface : Strategies for Effective Human-Computer-Interaction.* Addison Wesley Longman.

Self J., "Formal Approaches to Student Modeling", in McCalla G.I., Greer J. (Eds.). *Student Modeling: the Key to Individualized Knowledge-based Instruction.* Berlin: Springer-Verlag. 195-352.

Siegel J. 1996. *CORBA Fundamentals and Programming.* John Wiley & Son.

Song J. S., Hahn S. H. 1997. "Intelligent Tutoring System for Introductory C Language Course". *Computers and Education.* 28(2): 93-102.